MULTIPLE OUTLIERS IN LINEAR REGRESSION: ADVANCES IN DETECTION

METHODS, ROBUST ESTIMATION, AND VARIABLE SELECTION

by

James Walter Wisnowski

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

May 1999

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30.Jul.99	DISSERTATION

4. TITLE AND SUBTITLE
MULTIPLE OUTLIERS IN LINEAR REGRESSION: ADVANCES IN DETECTION MEHODS, ROBUST ESTIMATION, AND VARIABLE SELECTION

5. FUNDING NUMBERS

6. AUTHOR(S)
CAPT WISNOWSKI JAMES W

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ARIZONA STATE UNIVERSITY

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

FY99-246

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

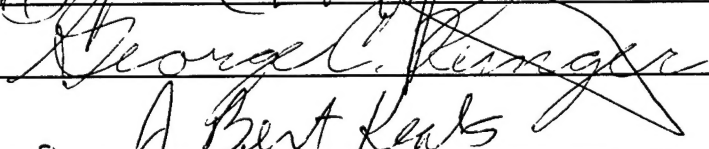# MULTIPLE OUTLIERS IN LINEAR REGRESSION: ADVANCES IN DETECTION
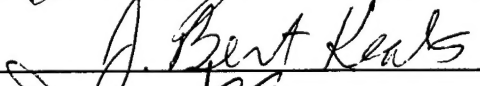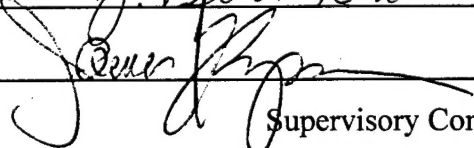
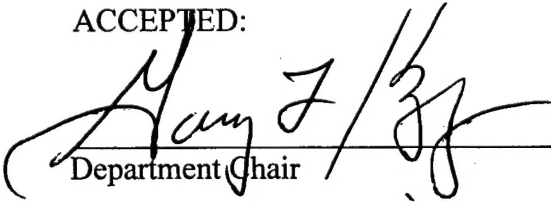# METHODS, ROBUST ESTIMATION, AND VARIABLE SELECTION

by

James Walter Wisnowski

has been approved

May 1999

APPROVED:

_____ ,Co-Chair

_____ ,Co-Chair

_____

_____

Supervisory Committee

ACCEPTED:

_____
Department Chair

_____
Dean, Graduate College

ABSTRACT

Empirical evidence suggests unusual or outlying observations in data sets are much more prevalent than one might expect; 5 to 10% on average for many industries. This research addresses multiple outliers in the linear regression model. Although reliable for a single or a few outliers, standard diagnostic techniques from an ordinary least squares (OLS) fit can fail to identify multiple outliers. The parameter estimates, diagnostic quantities and model inferences from the contaminated data set can be significantly different from those obtained with the clean data. The researcher requires a dependable method to identify and accommodate these multiple outliers.

This research tests both direct methods from algorithms and indirect methods from robust regression estimators to identify multiple outliers. A comprehensive Monte Carlo simulation study evaluates the impact that outlier density and geometry, regressor variable dimension, and outlying distance have on numerous published methods. The performance study focuses on outlier configurations likely to be encountered in practice and uses a designed experiment approach. The results for each scenario provide insight and limitations in performance for each technique. Recommendations are given for each technique.

OLS is the optimal regression estimator under a set of assumptions on the distribution of the error term and predictor variables. Compound robust regression estimators have been proposed as alternatives when some OLS assumptions fail. Compound estimators can accommodate multiple outliers and limit the influence of the observations with remote levels of predictor variables. This research proposes a new

compound estimator that is more effective for extreme observations in X-space and high-dimension than currently published methods.

This research also addresses the variable selection problem for compound robust regression estimators. Estimating model prediction error with resampling methods (bootstrap and cross-validation) is the most effective approach to the variable selection problem in OLS. Current research suggests that the best method for variable selection is to select the model with the minimum value of prediction error from a modified bootstrap procedure. The modified procedure uses a bootstrap sample size significantly less than the original sample size. A selection criterion is proposed based on a low prediction error (not necessarily minimum) with the fewest predictor variables. The proposed criterion often provides superior results to the minimum prediction error criterion and does not require the modified bootstrap procedure to achieve good results in OLS. Monte Carlo simulation results suggest that the proposed criterion is also effective for compound estimators in contaminated samples. This research shows the viability of combining the two computationally intense procedures of resampling methods and compound estimation to achieve accurate model selection in the presence of multiple outliers.

ACKNOWLEDGEMENTS

and my wife convinced me to apply for this outstanding opportunity. I truly value their constant support and admire their love of life.

My parents deserve special recognition. I greatly appreciate their love and support throughout my life. They have provided the character and self-discipline that has allowed me to persevere.

I am most grateful to my beautiful wife Shelley for believing in me and filling in the family voids when Daddy was being a "boxhead". My children, Chad and Chase, are wonderful diversions that helped me keep everything in perspective. For now, we are anxious to head back to the high country, but the great state Arizona has not seen the last of us.

TABLE OF CONTENTS

xi

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background and Motivation for this Research

The goal of the field of statistics is to transform raw data into useful information for decision making. By its nature, statistics is not an exact science and an approximation to an underlying process is often based on a sample of observations from the total population of interest. A common objective in statistics is to identify an appropriate transformation from a sample to relate a response (dependent) variable to a set of independent variables. Linear regression is the customary method used to mathematically model a response variable as a function of the regressor (independent) variables. Regression analysis is used in all fields of engineering, science, and management. Proliferation of the method continues because common software packages include regression options.

The regression model for n observations and k regressor variables can be described in terms matrices as $y = X\beta + \varepsilon$ where y is the $n \times 1$ vector of observed response values and X is the observed $n \times p$ matrix of k regressor variables augmented with a column of ones. β is an unknown $p \times 1$ vector of regression coefficients and ε is the $n \times 1$ vector of error terms. The ε vector is critical. If it is identically 0, then the process modeled is deterministic (e.g. $F=ma$). However, in practice, it is not identically 0 and the relationship between the response and predictor variables is not exact. That is, given the same set of regressor variables, the response values will not necessarily be the

same. The goal of regression analysis is to find a good estimate of the unknown regression coefficients β from the observed sample.

The usual estimator of β comes from the method of ordinary least squares (OLS) discovered independently by Gauss in 1795 and Legendre in 1805. OLS minimizes the sum of the squared distances for all points from the actual observation to the regression surface. The least squares estimator is attractive because of computational simplicity, availability of software, and statistical optimality properties. From the Gauss-Markov theorem, least squares is always the best linear unbiased estimator (BLUE). BLUE means that among all unbiased estimators, OLS has the minimum variance. If ϵ is assumed to be normally, independently distributed with mean 0 and variance $\sigma^2 I$, least squares is the uniformly minimum variance unbiased estimator. Under this assumption, inference procedures such as hypothesis tests, confidence intervals, and prediction intervals are powerful. However, if ϵ is not normally distributed, then the OLS parameter estimates and inferences can be flawed.

Violation of the NID distribution of the error term can occur when there are one or more outliers in the data set. An outlier is an observation that is inconsistent with the remainder of the data and it is not unusual to see an average of 10% outliers in data sets for some processes (Barnett and Lewis, 1994 and Hampel et al., 1986). Some of the sources of outliers are errors in data entry or measurement, the inadvertent inclusion of an observation from another population or a plausible event.

To illustrate the effect on least squares regression of an outlier, consider the pilot plant data from Daniel and Wood (1971) where the extraction rate is thought to be a

predictor of the acid content measured by titration. The two fits displayed in Figure 1.1 are of the original data and a hypothetical situation where a single transcription error is made on an extraction rate that changes it from 37 to 370 (Rousseeuw and Leroy, 1987). The correct least squares fit is $\hat{y} = 35.5 + 0.32x$ where $\hat{y}$ is the expected response value of y conditional on the level of x. The intercept and slope estimates for the model fit with the outlier are distinctly different, $\hat{y} = 58.9 + 0.08x$. Unless interest is confined to the region around mean level of the extraction rate, the outlier-contaminated model is poor. Both of the parameter estimates (intercept and slope) have changed too much from the true underlying relationship to be considered a meaningful description for the majority of the data. The OLS estimates have broken down after a single anomalous observation. Breakdown is a critical concept for this research. Huber's (1981) operational definition is the smallest fraction of data contamination needed to cause an arbitrarily large change in the parameter estimates.

Figure 1.1 clearly indicates that it is of interest to the regression practitioner to have a set of reliable tools to detect outlying observations. Fortunately, isolating a single or a few outliers in OLS is relatively easy with routine diagnostics (e.g. Cook's D, DFFITS, scaled residuals and residual graphics) supplied by most statistical analysis software. Multiple outliers in a sample have a similar, if not worse, effect on the least squares parameter estimates and inference as displayed in Figure 1.1. However, the standard diagnostic measures often fail to indicate anything unusual about these
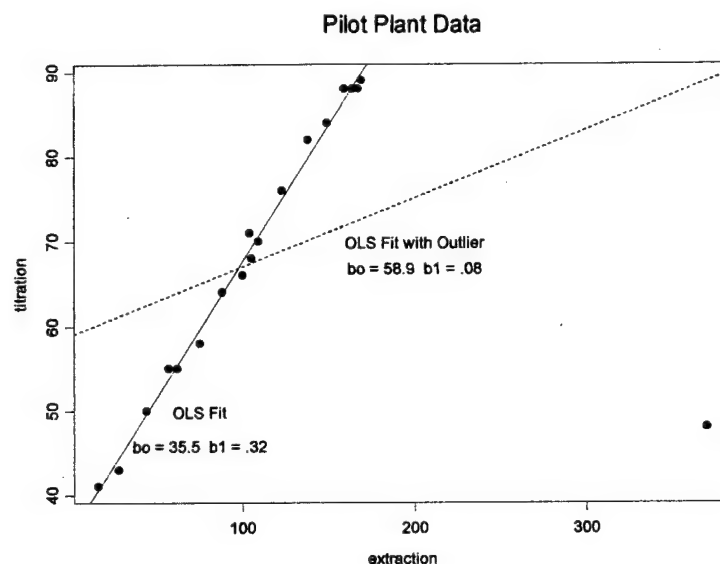
Pilot Plant Data

Figure 1.1. Breakdown of the ordinary least squares (OLS) estimator
in the modified pilot plant data (Daniel and Wood, 1971).

observations. Also, these diagnostics can signal that clean observations are outliers. The

former symptom of multiple outliers is known as masking and the latter is termed

swamping. There are several methods proposed in the literature that attack the multiple

outlier identification problem; yet, there is little guidance for the practitioner on which

methods perform well in representative outlier scenarios. Few methods are readily

available on standard statistical packages.

If the multiple outliers are successfully identified, a decision has to be made on

what to do with them. These aberrant observations could be left in the analysis. Figure

1.1 graphically depicts the consequences of leaving an outlier in the analysis.

Conversely, the outliers could be removed entirely from the analysis. If the outliers are

plausible events, then these observations may be the most important ones in the sample.

Dismissal of these outliers from the analysis could be a missed opportunity to characterize the process at certain operating conditions. A compromise between including and deleting the outliers is to downweight their influence on the regression surface. Robust regression estimators have been proposed as alternatives to OLS to downweight observations as a function of "outlyingness" in parameter estimation.

There has been a large body of literature in recent years developing the theory and practice of robust regression estimators. Typically, these estimators require significant computational resources because of nonlinear solutions or the requirement to search numerous subsets of the data to satisfy a constrained objective function. Ironically, the first robust regression estimator pre-dates OLS by nearly a half century. The L_1 or least absolute value estimator (Boscovich, 1757) is particularly well-suited for those heavy-tailed distributions (e.g. double exponential) that can generate outliers. However, this and many other robust regression estimators are not able to accommodate multiple outliers. That is, they are not high-breakdown estimators and fail with only a modest amount of outliers. The most often used high-breakdown estimators are the Least Median of Squares (Rousseeuw, 1983), Least Trimmed Sum of Squares (Rousseeuw, 1984) and S-estimators (Rousseeuw and Yohai, 1984). The problem with these estimators is that they can fail if the outliers have extreme values in the regressor variables (high-leverage points). The ability of a robust estimator to accommodate high-leverage outliers is called bounded-influence. The LMS, LTS, and S-estimators are also not efficient estimators. They do not fit data sets particularly well when there are no outliers present. Recently, a class of robust regression estimators has been proposed that

simultaneously achieve all three properties (Simpson, et al. 1992, Coakley and Hettmansperger, 1993, and Simpson and Montgomery, 1998). These compound estimators have the potential not only to identify a wide range of multiple outlier configurations, but also to accommodate them in a model. Hampel (1997) recommends such an approach to make the robust regression and regression diagnostic fields complementary rather than antagonistic.

Therefore, one method to detect multiple outliers in regression is to examine the final weights (between 0 and 1) that the robust regression estimator assigns each observation. Observations with weights close to 0 are candidates for outliers. Residual values from a robust fit can also be used to identify the outliers. There are also more direct multiple outlier detection procedures in the literature that use specially designed algorithms. There is little guidance and few empirical studies on which methods work best.

A tacit assumption to this point is that the correct regressor variables are specified for the model. Most regression modeling requires selection of the subset of regressor variables from a larger pool thought to be related to the response. Outliers confound the variable selection process because a variable that truly has no effect on the response may appear to be significant because it is fitting the outliers. Equally troublesome, the outliers may mask a significant variable. As an example, consider the modified Gunst and Mason (1980) data set created in Section 5.6.2 that has $n = 40$ observations, $k = 4$ regressor variables and four outliers. The OLS parameter estimates along with those from the proposed compound estimator in this research are displayed in Table 1.1. For this data

set, it is known $\beta' = [2, 0, 0, 4, 8]$. Least squares has fit the outliers with the two inactive regressor variables x_1 and x_2. Note that the compound estimator is resistant to the outliers.

Table 1.1. Least squares and proposed compound estimates of the regression parameters in the modified Gunst and Mason (1980) data.

	β_0	β_1	β_2	β_3	β_4
true parameter	2.00	0.00	0.00	4.00	8.00
least squares estimate	1.71	9.85	7.75	-2.62	10.17
compound estimate	2.24	-0.23	0.61	3.35	8.23

Selection of the best subset of variables is a critical part of the regression model building process. Again, there are numerous methods and criteria available to the practitioner with many accessible in standard statistical analysis software. Recent developments have suggested that resampling methods are better suited for the variable selection problem (Breiman, 1995, Shao, 1996, and Davison and Hinkley, 1997). It is not known how resampling methods perform with multiple outliers in the data.

1.2 Statement of the Problem

This research was introduced by defining the goal of the field of statistics. Staudte and Sheather (1990) claim that a better description with respect to robust estimation may be the "battlefield of statistics" because of the controversy surrounding many of the proposed techniques. Most of the community does agree that there is no single best robust regression estimator, multiple outlier identification procedure or

variable selection procedure. However, there are widely varying opinions as to the applicability of certain methods in specific scenarios.

Hettmansperger (1998) states one of the main reasons why robust estimation is not used more in statistics is the "curse of abundancy" for the techniques. His point is that not only are there many different estimators and algorithms available, but also that each procedure has its own, often large, set of parameter settings and tuning constants. Hettmansperger also states the lack of software for robust procedures is another reason attributing to the scarcity of robust analysis.

These two reasons present a challenging dichotomy to the regression user. On the one hand, extra effort is often required to get the appropriate software to implement existing robust procedures. However, once software capability is achieved, the practitioner is saturated with implementation options. Performance studies are needed in finite samples to screen many of the existing procedures and quantify where each is best suited.

To this end, a comparison of multiple outlier detection procedures across a comprehensive set of scenarios is missing in the literature. The ideal outcome of such a study would be that one procedure is preferred in all scenarios. If this is not the case, characterization of effective areas of technique performance would be helpful guidance. The comparative evaluation could also suggest that some techniques could be improved to make them "robust" to more outlier scenarios. A similar approach has been taken by Simpson and Montgomery (1998c) to propose a "robust" robust regression estimator.

There also is no shortage of options for variable selection in regression. Some performance studies exist and resampling methods are preferred. No work in the literature addresses the combined variable selection in the presence of outliers with compound estimators. This is understandable because compound estimation is highly computer intensive and resampling methods increase complexity by orders of magnitude.

1.3 Research Objectives

There are three primary objectives that address the research problem.

- Characterize the performance of the leading multiple outlier detection procedures for the linear regression model. The goal is a comprehensive evaluation of published techniques that suggests where the methods are successful and where they fail. A successful procedure would have a high probability of identifying the outliers, a low probability of classifying clean observations as outliers and be easily implemented in analysis software.

- Select and improve upon the most promising techniques from the comparative study of multiple outlier detection procedures. This phase could improve upon a direct multiple outlier detection algorithm, a robust regression estimator, or a combination both.

- Determine the appropriateness and computational feasibility of resampling methods for variable selection in the presence of outliers. This phase specifically addresses variable selection with compound robust regression estimators using the bootstrap and cross-validation procedures.

1.4 Scope of Research

The research focus is on the finite sample size performance of the published techniques and also those proposed in this research. The selection of the techniques is limited to those that are promising and often referenced in the literature and those that perform well in pilot studies. For this research, the problems of outlier identification, robust estimation and variable selection are limited to the linear regression model. Nonparametric, Bayesian, and nonlinear regression and generalized linear models are not considered; although, many of the concepts explored easily extend to those classes of models.

Monte Carlo simulation is the primary tool to accomplish the objectives outlined in Section 1.3. This computer-implemented technique generates numerous data sets by randomly varying specific values at each iteration. For comprehensive test and evaluation of the methods, it is not possible to cover many of the infinite factor levels that characterize a data set such as the number of observations (n), number of regressors (k), percentage of outliers, outlier location, and magnitude of outliers. Representative and interesting levels of these factors are selected from pilot studies and sequential analysis. Additionally, each individual technique has its own set of specific parameter settings. Either the default or most favorable settings from pilot studies are used. In most cases, the Monte Carlo simulation experiments are set up as factorial designs to gain the maximum understanding from a moderate amount of experimentation. Although many references suggest using thousands of Monte Carlo simulation replicates, the nature of

this problem does not lend itself to such luxury. In all cases, there are enough replicates to get a clear indication of performance.

1.5 Summary and Outline of Research

The goal of this research is to comprehensively and fairly evaluate the leading candidate multiple outlier detection procedures. These results are then used to introduce an improved method. Variable selection for compound estimators is then considered with resampling methods.

Chapter 2 reviews the relevant literature on what has been published to date. Chapters 3 through Chapter 5 are essentially stand-alone documents that address each of the research objectives. Chapter 3 details the Monte Carlo simulation performance study of multiple outlier detection procedures. Chapter 4 proposes a new compound estimator using results from extensive performance studies on measures of leverage and high-breakdown estimators. Chapter 5 addresses the variable selection problem in linear and robust regression using resampling methods. A new variable selection criterion is introduced that proves effective with both least squares and compound estimators. Chapter 6 provides a summary of the results, the contributions, and the recommendations for further research.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the related literature for this research. Chapters 3, 4 and 5 are written as larger versions of what is to be submitted for publication. As such, each chapter contains its own literature review restating many of the results presented here. The difference is that more explanation of the key concepts and algorithms is treated in Chapter 2. This review begins with some background material on least squares regression estimation and diagnostics to identify a single outlier. The discussion expands to address the multiple outlier problem. A detailed presentation of direct procedures to identify multiple outliers is followed by a thorough discussion of the indirect identification procedures; namely robust regression estimators. The chapter concludes with a discussion of variable selection methods in linear regression.

2.2 Ordinary Least Squares Regression

Regression analysis models the relationship between a response variable and a set of predictor variables. The regression model for n observations and k regressor variables can be described in terms of matrices as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where $\mathbf{y}$ is the $n \times 1$ vector of observed response values and $\mathbf{X}$ is the observed $n \times p$ matrix of k regressor variables augmented with a column of ones. $\boldsymbol{\beta}$ is an unknown $p \times 1$ vector of the regression coefficients and $\boldsymbol{\varepsilon}$ is the $n \times 1$ vector of error terms. In practice, the regression model is

$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} + \mathbf{e}$ where $\hat{\mathbf{y}}$ is the vector of predicted response values, $\mathbf{e}$ is the vector of residuals, and $\hat{\beta}$ the estimate of regression coefficient. OLS computes these parameter estimates, $\hat{\beta}$, by minimizing the sum of the squared residuals. Therefore, the objective is to find those values of $\hat{\beta}$ that lead to the minimum value of $\mathbf{e}'\mathbf{e} = \sum_{i=1}^{n} e_i^2$. Nearly all regression texts (e.g. Montgomery and Peck, 1992) give the fundamental derivation of the OLS parameter estimates as follows:

$$
\begin{aligned}
S(\beta) \quad = \varepsilon'\varepsilon \;\; &= (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) \\
&= \mathbf{y}'\mathbf{y} - \beta'\mathbf{X}'\mathbf{y} - \mathbf{y}'\mathbf{X}\beta + \beta'\mathbf{X}'\mathbf{X}\beta \\
&= \mathbf{y}'\mathbf{y} - 2\beta'\mathbf{X}'\mathbf{y} + \beta'\mathbf{X}'\mathbf{X}\beta
\end{aligned}
$$

using differentiation to minimize $S(\beta)$ with respect to β,

$$
\left. \frac{\partial S}{\partial \beta} \right|_{\hat{\beta}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{0}
$$

rewriting and simplifying gives the least squares normal equations

$$
\mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{X}'\mathbf{y}
$$

which can be solved if $\mathbf{X}'\mathbf{X}$ is of full rank for the familiar OLS relationship

$$
\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}
$$

The vector of fitted values can be expressed as

$$
\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}\mathbf{y}
$$

The matrix **H** is known as the hat or projection matrix. The diagonal elements of the hat matrix are used in many least squares diagnostics because they provide an indication of remoteness in X-space.

Some useful properties of $\hat{\beta}$ are that it is an unbiased estimator ($E(\hat{\beta}) = \beta$) and the Gauss-Markov theorem guarantees that among all unbiased estimators of β, the least squares estimate has the minimum variance, $Cov(\hat{\beta}) = \hat{\sigma}^2 (X'X)^{-1}$. A common estimate for σ^2 is the Mean Square Error (MS_E) = $e'e/(n-p)$. The least squares estimator of β is also the maximum likelihood estimator under the assumption that the error terms are independent and identically distributed normal variates with mean **0** and covariance matrix $\sigma^2 I$. The usual notation for this assumption is $\varepsilon \sim NID(0, \sigma^2 I)$ and if it holds, then OLS is also the uniformly minimum variance unbiased estimate (UMVUE). Model inferences such as confidence intervals and hypothesis tests are also very powerful if the error terms are NID. From a statistical point of view, OLS is the optimal estimator under a normal error.

The major disadvantage of OLS is performance when the error term cannot be assumed to be distributed normal. OLS estimates and tests rapidly lose power with nonnormal error terms. One of the most common violations of a normal distribution for the error terms is the presence of one or more outliers in the sample.

2.3 An Outlier in Least Squares Regression

Barnett and Lewis (1994) define an outlier as an observation that appears inconsistent with the remainder of the data set. Outlier identification is important in OLS

due not only to their impact on the OLS model, but also to provide insight into the process. These outlying cases may arise from a different distribution altogether from the bulk of the data. The distribution of the full dataset is contaminated in this instance. In contaminated datasets, it makes sense to see if there may be an alternative model form (e.g. lognormal as opposed to normal errors) to fit the true process. Alternatively, the distribution of the unusual observations may be uncontaminated but there may have been an external cause such as a recording or interpretation error. These two cases may require a different approach on how to accommodate the outlier: include it in the model, downweight it and include it in the model, or throw it away.

To classify types of outliers for this research, consider the simple linear regression model displayed in Figure 2.1. The ellipse defines the majority of the data. Point A is an outlier in Y-space because its response value is significantly different from the responses contained in the ellipse. Point A is also a residual or regression outlier. Its expected response, conditional on the value of x, differs significantly from the regression line fitted to the data in the ellipse. Point B is unusual in X-space. Observations that are remote in X-space are high-leverage points and are also referred to as exterior X-space observations in this research. Although Point B is remote in Y-space, it is not a residual outlier because its response value conforms to the regression line fit to the observations in the ellipse. Point B can be considered a "good outlier". Points C and D are high-leverage points and residual outliers. Point C is unusual in Y-space; point D is not.
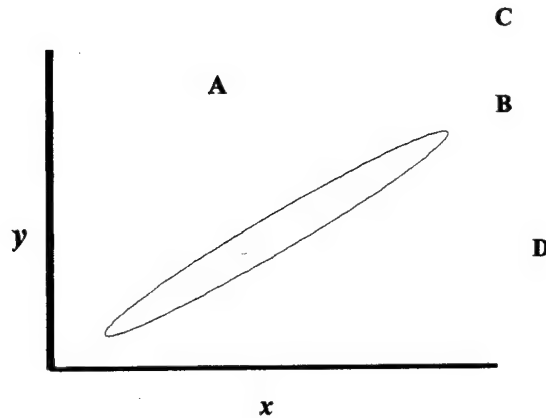
Figure 2.1. Outlier configurations: Points A, B, and C are outlying in Y-space (exterior Y-space), Points B, C, and D are high-leverage points (exterior X-space), Points A, C, and D are residual outliers because they do not conform to the regression line defined by the clean observations in the ellipse.

2.3.1 Detection of an Outlier in X-space

The effect of outliers in X or XY-space is to "pull" or exert more influence on the model parameters estimating the regression line. These observations are influential or high-leverage points. When there are three or fewer regressor variables, candidate outlying observations in X-space can be detected by a three or two-dimensional scatterplot of the regressor variables. Computational measures are needed for more than three variables. The diagonals, h_{ii}, of the $n \times n$ hat matrix, $\mathbf{H}$, provide a measure of remoteness in X-space. Because the sum of the hat diagonals is p, the average of all the hat diagonals is p/n. Hoaglin and Welsch (1978) suggest observations with h_{ii} greater than 2 or $3p/n$ should be considered as potential outliers.

A related measure for multivariate distance in X-space is the Mahalanobis distance and is defined for the ith observation as:

$$D_i^2 = (\mathbf{x}_i - \mu)'\Sigma^{-1}(\mathbf{x}_i - \mu)$$

where $\mathbf{x}_i$ is the $p \times 1$ vector of regressor variables for case i, μ is the mean vector of $\mathbf{X}$ and Σ is the $p \times p$ sample covariance matrix. If the classical estimates of μ and Σ from the full sample are used, then it can be shown that the hat diagonal is a function of the Mahalanobis distance, $h_{ii} = \dfrac{D_i^2}{n-1} + \dfrac{1}{n}$. Observations with D_i greater than $\chi^2_{(p,1-\alpha/2)}$ are potential outliers (Rousseeuw and van Zomeren, 1990).

2.3.2 Detection of a Residual Outlier

An observation with a relatively large residual value is a candidate for an outlier. Scaling each residual, e_i, can often help detect outliers. Standardized residuals correct for the overall model variance and are calculated for each observation as $d_i = \dfrac{e_i}{\sqrt{MS_E}}$.

Under the assumption of normally distributed error terms the standardized residuals can be compared to the percentiles of the standard normal. A possible problem with this approach is that the variance of the residuals depends on their location in X-space, $\mathrm{Var}(\varepsilon_i) = \sigma^2(1-h_{ii})$. Behnken and Draper (1972) suggest the constant variance studentized residual defined as $r_i = \dfrac{e_i}{\sqrt{MS_E(1-h_{ii})}}$. The studentized deleted residual replaces MS_E in the previous equation with the variance estimate obtained by removal of the ith observation as $s^2_{(i)} = \dfrac{(n-p)MS_E - e_i^2/(1-h_{ii})}{n-p-1}$. Montgomery and Peck (1992) state that the studentized deleted residual is preferred in dealing with outliers especially since it follows the t-distribution. Allen (1971) states that the residual obtained by using

a model fitted with a sample that omits the observation is the prediction error sum of squares residual (PRESS). The PRESS residual is easily calculated in least squares as

$$e_{(i)} = \frac{e_i}{(1 - h_{ii})}$$ and does not require n separate OLS fits.

2.3.3 Influence Measures in Least Squares Regression

The hat diagonal and residual measures are useful diagnostic measures to quantify an observation's remoteness in X-space and the distance off the regression surface. However, they do not provide an indication of how the model parameter estimates or fitted values are impacted by inclusion of the potential outlier. Influence diagnostic measures have been developed to help in making the decision of what to do with an unusual observation. That is, an observation may be a high-leverage point and residual outlier, yet inclusion in the analysis has little effect on model parameter estimates and inferences. Barrett and Gray (1997) state most influence diagnostics can be decomposed into a measure of leverage and a measure of residual.

Cook's Distance (Cook, 1979), D_i, incorporates both the remoteness in X-space and in residual.

$$\text{Cook's } D_i = \frac{\left(\hat{\beta}_i - \hat{\beta}_{(i)}\right)' \mathbf{X}'\mathbf{X}\left(\hat{\beta}_i - \hat{\beta}_{(i)}\right)}{p MS_E} = \frac{r_i^2}{p}\left(\frac{h_{ii}}{1 - h_{ii}}\right)$$

where $\hat{\beta}_{(i)}$ is the vector of parameter estimates from the OLS fit with observation i and r_i^2 is the squared studentized residual. Cook recommends that distances greater than $F_{\alpha=.5,p,n-p} \cong 1.0$ are considered influential.

Belsley, Kuh, and Welsch's (1980) *DFBETAS* statistic considers the impact of leverage and residual on each of the p parameter estimates. The statistic measures how each parameter estimate changes if the ith observation is removed from the data set. Observations with *DFBETAS* exceeding $2/\sqrt{n}$ in magnitude are influential.

$$DFBETAS_{j,i} = \frac{\hat{\beta}_j - \hat{\beta}_{j,(i)}}{\sqrt{S^2_{(i)}C_{jj}}}$$

where $\hat{\beta}_{j,(i)}$ is the OLS estimate of the jth regression coefficient from a fit without the ith observation and C_{jj} is the jth diagonal element of $(\mathbf{X'X})^{-1}$.

Belsley, Kuh and Welsch (1980) introduced *DFFITS* to measure the influence on the predicted values by omission of the ith observation. Observations exceeding $2\sqrt{p/n}$ in absolute value are considered influential.

$$DFFITS_i = \frac{\hat{y}_i - \hat{y}_{(i)}}{\sqrt{S^2_{(i)}h_{ii}}}$$

Belsley, Kuh and Welsch (1980) define the COVRATIO statistic to measure the overall precision of estimation. This statistic is based on the ratio of generalized variances found from the determinant of the covariance matrix.

$$COVRATIO_i = \frac{|(\mathbf{X'}_{(i)}\mathbf{X}_{(i)})^{-1}S^2_{(i)}|}{|(\mathbf{X'X})^{-1}MS_E|} = \frac{\left(S^2_{(i)}\right)^p}{MS_E^p}\left(\frac{1}{1-h_{ii}}\right)$$

Belsley, Kuh, and Welsch suggest that observations varying by more than $3p/n$ from unity may be influential.

The leverage, residual, and influence measures described above can effectively isolate an outlier and provide an indication as to how much it is affecting the model. Many authors recommend complementing these procedures with a plot of predicted versus residual values, a normal probability plot of residuals, and plots of each regressor versus residual values for outlier detection. Cook (1998) gives guidance on numerous other modern graphical procedures that can provide insight into outliers and influence in regression. The problem with these quantitative and graphical approaches to the outlier problem is that they can fail if there are multiple outliers. Kempthorne and Mendel (1990) discuss the inadequacies of these single row influence diagnostics when applied to multiple observations.

2.4 Detection of Multiple Outliers with Direct Methods

The reason many of the techniques in Sections 2.3.2 and 2.3.3 fail with multiple outliers is that they are a function of the covariance matrix. If there are too many outliers, then the estimate of the covariance matrix is poor and biased toward the outliers. The two primary symptoms from multiple outliers in regression are masking and swamping. Masking occurs when the true outliers are not identified. This inflates the estimate of error thereby affecting the power of test statistics. Swamping occurs when inliers are identified as outliers. One possible solution to the problem is to analyze subsets of the observations thought to be outliers.

Belsley, Kuh and Welsch (1980) and Cook and Weisberg (1982) extend their influence diagnostic measures to accommodate subsets of observations rather than just

one. These authors and Sebert (1995) demonstrate that the multiple row diagnostics effectively assess the joint influences exerted by several outliers. Barrett and Ling (1992) and Barrett and Gray (1997) propose improved multiple row diagnostics that are based on measures of leverage, residual and the interaction between the two. The problem with all multiple row influence diagnostics is that the correct subset must be tested. This presents a significant combinatorial problem with increasing sample size. Several procedures to identify this outlying set have been published in the last 25 years. Hadi and Simonoff (1993) classify the procedures as direct or indirect. Direct procedures use a specifically designed algorithm to detect multiple outliers. The indirect methods use either the weights assigned to each observation or the residuals from a fit with a robust regression estimator.

This section chronologically describes the direct methods to detect multiple outliers in linear regression. For the procedures that are used in the performance studies, there is a detailed outline of the algorithm that is significantly expanded from the short summary provided in chapter 3. There is a brief description of several other procedures for historical purposes and reference.

2.4.1 Gentleman and Wilk Subsets Algorithm

Gentleman and Wilk (1975) are generally credited with first addressing methods to detect the multiple outliers in the least squares regression model. Their Q statistic is based on the reduction in error sum of squares from the model including all observations to that of a model of size $(n - j)$ where j is the pre-specified maximum number of potential

outlying cases. This statistic is computed for all subsets of size j and those subsets with large Q values are considered potential outlier sets. The method's limitations are the computational complexities for large n and the requirement to specify the expected number of outliers. Gentleman (1980) addressed the computational complexity issue by sequentially selecting the set of outliers based on OLS studentized residuals from the full sample. This procedure still suffers from masking and swamping because studentized residuals may not appear unusual if there are multiple outliers.

2.4.2 Hawkins, Bradu, and Kass Elemental Sets Algorithm

Hawkins, Bradu and Kass (1984) identify multiple outliers in regression models with elemental sets. Numerous random samples of size p are formed from the original data set and fit with an OLS regression model. Outliers are the observations with large values for the summary statistics (e.g. median) on the set of residuals from all of these regressions. This procedure is similar to bootstrapping and suffers from computational complexity. Also, the OLS residuals may not be unusual for the high-leverage regression outliers.

2.4.3 Marasinghe Backward Selection Algorithm

Marasinghe (1985) proposes a multi-stage procedure that also requires specification of the expected maximum number of outliers. The outliers are sequentially removed based on the largest absolute value of the studentized residual. The test statistic F_j is the ratio of error sum of squares from the reduced model with $(n - j)$ observations to

the error sum of squares for the full sample model. If F_j exceeds a critical value from the Bonferroni inequalities, then the current set is the outlier set; otherwise, the procedure is repeated using $(j - 1)$ candidate outliers. This methodology again relies on the studentized residual which suffers from masking and swamping; particularly if j is specified too large (Fung, 1988).

A proposed methodology by Kianifard and Swallow (1989, 1990 and described for their 1996 update in Section 2.4.9) was compared to Marasinghe (1985) and modifications to Gentleman and Wilk (1975) in several outlying scenarios. The results were scenario dependent; however, for multiple outliers, Marasinghe's multi-stage procedure performed the best. Kianifard and Swallow also note the poor performance from misspecification of the number of outliers in Marasinghe's procedure.

2.4.4 Rousseeuw and van Zomeren MVE/LMS Plot

The highly-referenced Rousseeuw and van Zomeren (1990) methodology is based on robust regression estimators. They suggest using the minimum volume ellipsoid (MVE) described in Section 2.5.4 as a robust estimate of both the mean and covariance matrix to detect outliers in X-space. The robust distance from the Mahalanobis Distance using the MVE estimates of the mean and covariance matrix can be compared to the $\chi^2_{p-1,0.975}$ distribution to conclude whether the point is influential in X-space only. The standardized residuals from a least median of squares fit (see Section 2.5.2) are used to identify residual outliers. This procedure then classifies an observation into one of four categories: 1) not an outlier, 2) a residual outlier only, 3) a leverage outlier, or 4) an

outlier in both residual and X-space. The methodology performs well on several challenging multiple outlier data sets where the classical distance measures fail. This method is available in the *S-Plus* software.

This procedure, as Cook and Hawkins (1990) noted in their discussion of the paper, suffers from identifying too many outliers. Other authors have similar reservations about using the MVE (Simpson, 1995, and Woodruff and Rocke, 1994). However, there have been improvements to the MVE and LMS algorithms recently that increase the statistical and computational efficiencies of the procedures (Burns, 1992).

2.4.5 Paul and Fung Backward Selection Algorithm

The Paul and Fung (1991) two-phase procedure using generalized extreme studentized residuals (GESR) tries to minimize the effect of overspecifying j, the maximum number of outliers, in the Marasinghe (1985) method. The algorithm forms the set of up to j residual outliers by sequential deletion of the observation with the highest absolute value of the studentized residual. This residual value must exceed a Bonferroni critical value. A model is refit without the potential outliers and the largest studentized residuals are tested again. A similar procedure used in phase two is to search for outlying values in X-space using Cook's D. The union of these two sets would be declared the outliers. Hadi and Simonoff (1993) show through benchmark examples and simulation that this method suffers from both masking and swamping.

2.4.6 Hadi and Simonoff Forward Selection Algorithm

Hadi and Simonoff (1993) consider two related procedures for the identification of multiple outliers in regression models. Their procedure is based on finding a "clean" subset of $n - k + 1$ observations that has the minimum residual sum of squared errors. The algorithm proceeds as:

1. Determine the initial clean subset M of size $h = (n + k - 1)/2$.

   a. Version 1 for determining M is an adaptation of Hadi (1992, 1994) and related to the elemental sets of Hawkins, Bradu, and Kass (1984).

   a.1 Order the n observations by the magnitude of the OLS adjusted residuals, $a_i = e_i / \sqrt{1 - h_{ii}}$.

   a.2 Form the basic subset B by selecting the $k + 1$ lowest values of the $|a_i|$.

   a.3 Fit an OLS model to set B

   Order the scaled residuals defined by

   $$sr_i = \frac{|y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_B|}{\sqrt{(1 - \mathbf{x}_i'\ (\mathbf{X}_B'\mathbf{X}_B)^{-1}\mathbf{x}_i}}, i \in B$$

   $$sr_i = \frac{|y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_B|}{\sqrt{(1 + \mathbf{x}_i\ (\mathbf{X}_B'\mathbf{X}_B)^{-1}\mathbf{x}_i}}, i \notin B.$$

   a.4 If s, the size of the basic subset, is equal to h then go to step 2, else use the first $s + 1$ observations ordered by the scaled residual as the new basic subset, B. Go to step a.3.

b. Version 2 for determining the initial clean subset M is based on Simonoff (1991) using a single linkage clustering algorithm to detect multivariate outliers.

b.1 Standardize the data by dividing $\mathbf{Z} = (\mathbf{X} : \mathbf{Y})$ by $\Sigma^{1/2}$. Note that the authors found the classical estimate of Σ superior to the MVE.

b.2 Construct the single linkage clustering tree for all n observations.

b.3 Order clusters from most to least extreme (the more extreme, the later the cluster joins) and consider cases in smaller clusters as potential outliers.

b.4 Cluster until there are $n - h$ "extreme" clusters; the remaining h cases are the clean data for the initial subset M.

2. Compute the internally studentized residual or scaled prediction error, d_i.

$$d_i = \frac{y_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}_M}{\hat{\sigma}_M \sqrt{1 - \mathbf{x}_i' (\mathbf{X}_M' \mathbf{X}_M)^{-1} \mathbf{x}_i}}, i \in M \ \text{(studentized residual)}$$

$$d_i = \frac{y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}_M}{\hat{\sigma}_M \sqrt{1 + \mathbf{x}_i' (\mathbf{X}_M \mathbf{X}_M)^{-1} \mathbf{x}_i}}, i \notin M \ \text{(scaled prediction error)}$$

3. Define s as the size of the current subset. If $|d_{(s+1)}| \geq t_{(\alpha/2(s+1), s-k)}$ then all observations with $|d_i|$ exceeding this critical value of the t distribution are outliers. Otherwise, find a new subset M by using the first $s + 1$ ordered observations. If $s +1 = n$, then there are no outliers to consider.

Hadi and Simonoff show that their methodology is successful in only two of the four "benchmark" example sets from the Rousseeuw and Leroy (1987) robust regression text. Kianifard and Swallow's (1989) procedure and Marasinghe's (1985) multi-staged approach fail on all four. The best procedure was the multi-staged robust regression *MM* estimator (see Section 2.5.3). Hadi and Simonoff also conduct a limited ($n=25$, $p=2$ or 3) Monte Carlo simulation using similar outlying scenarios to Kianifard and Swallow. The results suggest Least Median Squares, Reweighted Least Squares and Least Trimmed Sum of Squares perform poorly for outliers at low-leverage due to their low efficiency. A high-efficiency *MM* estimator is sensitive to high-leverage outliers and breaks down after 3 observations while a lower efficiency *MM* estimator (70%) is better for the high-leverage outliers at the expense of significant swamping. These results agree with Simpson's (1995) simulations where the only weakness for *MM* estimators is the combination of high-leverage and low-dimension. Hadi and Simonoff conclude that their procedure (Version 1) is preferable to all others based on computational ease, known cutoff values, and overall performance. Their estimators did not breakdown nor excessively swamp in the presence of multiple high-leverage points.

2.4.7 Atkinson Stalactite Plot

Atkinson (1994) uses a computationally attractive alternative to Rousseuuw and van Zomeren (1990) based on the LMS residuals and MVE. Forward selection minimizes the probability of including an outlier in the observations in the MVE. The search is conducted several times at random starting points to find the "global" MVE

subject to no outliers. Simulation is required to scale the LMS residuals to determine "outlyingness". Stalactite plots conveniently show which observations exceed a critical cutoff value from the scaled LMS residuals as a function of subsample size. When the size of the subsample is equal to the number of observations, the stalactite plot displays the effect of masking and offers guidance in selecting appropriate subsample sizes for protection against masking. The procedure performs well in several of the "benchmark" examples, but the algorithm is outdated for LMS and MVE.

2.4.8 Pena and Yohai Eigenanalysis

Pena and Yohai (1995) describe a procedure to detect influential subsets in regression using eigenanalysis on the influence matrix. The $n \times n$ influence matrix is defined as *the uncentered covariance of a set of vectors which represent the effect on the fit of the deletion of each data point.* Define $\mathbf{t}_i = \hat{\mathbf{y}} - \hat{\mathbf{y}}_{(i)} = \{e_i / (1 - h_{ii})\}\mathbf{h}_i$, where $\mathbf{h}_i$ is the ith column of the hat matrix. If $\mathbf{T} = (\mathbf{t}_1 \ldots \mathbf{t}_n)$, then the influence matrix $\mathbf{M} = \mathbf{T'}\mathbf{T}/ps^2$. The univariate Cook's Distance for each observation is on the diagonal of $\mathbf{M}$.

The algorithm is as follows:

1. Form the influence matrix as $\mathbf{M} = \mathbf{EDHDE}/ps^2$ where $\mathbf{E}$ is the diagonal matrix of residuals, $\mathbf{D}$ is the diagonal matrix with elements $(1 - h_{ii})^{-1}$, $\mathbf{H} = \mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'}$, and s^2 is the usual mean square error estimate of the variance.

2. Computationally, it is better to consider a decomposition of the influence matrix because only a subset of the eigenvectors is of interest. Define $\mathbf{A} =$

$\mathbf{B}\Lambda^{1/2}$ where the columns of $\mathbf{B}$ are the eigenvectors of $(\mathbf{X}'\mathbf{X})^{-1}$ and Λ is the diagonal matrix of the associated eigenvalues. If $\mathbf{P} = \mathbf{EDXA}/(p^{1/2}s)$, the eigenvectors from the non-null eigenvalues of the influence matrix $\mathbf{M}$ are $\mathbf{Pv}_i$ where $\mathbf{v}_i$ are the eigenvectors from $\mathbf{P'P}$.

3. Find the eigenvectors of the p non-null eigenvalues of the influence matrix.

4. Order the components within each eigenvector, $\mathbf{v}_i$, in ascending order to obtain the order statistics $v_{i(1)} \leq v_{i(2)} \leq \ldots v_{i(n)}$.

5. Search the eigenvectors for observations with large positive or large negative components. These sets will be considered candidates for outliers. The ratio $a_j = v_{i(j)}/v_{i(j-1)}$ for $j = n, \ldots, n - c_1$ searches for a breakpoint for the positive components. Similarly, $b_j = v_{i(j)}/v_{i(j+1)}$ for $j = 1, \ldots, c_2$ finds the breakpoint for the negative components. The constants c_1 and c_2 define what percentage of the total observations should be considered as potential outliers. In practice, the authors recommend $n/4$ for both values to detect up to 50% outlying observations. It also makes sense that both of these constants should be equal since we do not know a priori whether the outliers will load positively or negatively. In fact, experimentation in identical scenarios shows the outliers to load inconsistently between replicates.

6. Search for a possible negative and positive breakpoint in the components *in each eigenvector*. For a particular eigenvector, select the first j_0 such that $|a_j| \geq k$ then consider the observations corresponding to the jth ordered component up to n as candidate outliers. For the negative values on the components,

select the first j_o such that $|b_j| \geq k$ and consider the observations corresponding
to the largest negative ordered component up to the $j_o{}^{th}$ as candidate outliers.
The key to this step is selecting k, the minimum ratio required to declare
outliers. This parameter is highly significant in determining the tradeoff
between high power in detecting the outliers and high false alarm rate. The
authors recommend 2.5; however, this may lead to too many false alarms in
small samples.

7. The last step evaluates the candidate outlier sets identified from step 6 by
   eliminating the observations from an OLS fit and evaluating the t tests
   (Bonferroni) for each candidate outlier and outlier sets.

The authors' limited testing of the procedure shows it to perform well in high-
leverage cases, especially with a low amount of contamination. The method also
correctly identifies outliers from the challenging Hawkins, Bradu, and Kass dataset.

2.4.9 Swallow and Kianifard Recursive Residual Algorithm

Swallow and Kianifard (1996) address the deficiencies from their 1990 recursive
residual methodology for multiple outliers. The improved procedure replaces classical
estimates of variance with robust measures; the easily computed interquartile range (IR)
and median absolute deviation from the median (MAD). The IR estimate of the standard
deviation is the 75th percentile - 25th percentile of the OLS or recursive residuals. The
MAD estimate of standard deviation is median $\{|e_i - \text{median } \{e_i\}|\}$ using OLS or
recursive residuals.

The procedure works for both OLS and recursive residuals. In practice, the authors claim using recursive residuals almost always leads to greater detection power. The outlier detection algorithm using recursive residuals is as follows:

1. Order the studentized residuals from an OLS fit for all n observations.

2. Use the first p ordered observations for the basis for computing the recursive residual,

$$w_j = \frac{y_j - \mathbf{x}'\hat{\boldsymbol{\beta}}_{j-1}}{(1 + \mathbf{x}'_j \, (\mathbf{X}'_{j-1}\mathbf{X}_{j-1})^{-1}\mathbf{x}_j)^{1/2}}, j = p+1,...,n.$$

3. Compute the robust estimate of scale $\hat{\sigma}$. This is the MAD or IR using the OLS residuals divided by a correction factor. The correction factor is determined by finding the mean IR and MAD estimates from a simulation under the null hypothesis of no outliers with the same number of parameters and observations as the data being analyzed.

4. Compute the test statistics $\left|w_j / \hat{\sigma}\right|$ for each observation.

5. Compare the test statistics to a critical value. Again, the critical value must come from simulating the given scenario under the hypothesis of no outliers. The critical values are found as the quantiles of the distribution of test statistics. It turns out that the critical values are virtually identical whether using a MAD or IR estimate of scale. These values are also very similar to the percentiles of the standard normal density; particularly as n gets large.

6. If the test statistic exceeds the critical value, then classify the respective observation as an outlier. This is the "recursive method". The authors also

provide a modification that can be described as an inward procedure because

it looks at the maximum $\left| w_j / \hat{\sigma} \right|$ to see if the no outliers hypothesis can be

rejected. If it is rejected, then the observation with the largest test statistic is

declared an outlier, it is deleted and the procedure repeats itself by calculating

the new recursive residuals. This iterates until there are no outliers detected.

This version is extremely computer intensive and does not offer a significant

advantage over the basic recursive method.

Swallow and Kianifard run simulations with the same seven outlying scenarios as

their previous work (1990) and those of Hadi and Siminoff (1993) which are limited

because only a single regressor with $n = 25$ is used. The results show insensitivity to the

IR or MAD estimate of σ, no significant swamping in any scenario by any method,

moderately higher power in detecting outliers from recursive residuals over OLS

studentized residuals, and the usefulness of robust estimator over OLS in masking

scenarios. No method performs well until the outlying distance is at least 4σ. The

authors claim of simplicity as the primary advantage to their methodology is questionable

based on the number of simulations required for distribution properties.

2.4.10 Sebert, Montgomery, and Rollier Clustering Algorithm

Sebert et al. (1998) suggest an approach for identifying a reasonable candidate

subset of multiple outliers that avoids the complexities associated with most competing

procedures. The methodology clusters observations from an easily formed projection of

the data into two independent dimensions. Specifically, Sebert et al. suggest the following steps:

1. Standardize the predicted and residual values from an OLS fit.

2. Cluster these observations using Euclidean distance and a single linkage clustering algorithm.

3. Form clusters based on tree height (ch, a measure of closeness) using Mojena's stopping rule ($ch = \overline{h} + 1.25 s_h$ where $\overline{h}$ is the average height of the tree and s_h is the sample standard deviation of heights). Note that tree height is a measure of cluster separation.

4. The single largest cluster is the clean data while the remaining subsets are all candidates for outliers.

5. Assess the influence of the candidate observations using multiple row diagnostics.

Simulated regression data sets demonstrate the success of the methodology. The procedure is generally very powerful at detecting outliers and performs well on the classic challenging data sets. Other significant simulation results include:

- The correct observations are identified as outliers increasingly better as outlying distance, number of observations, number of regressors, and percentage of outliers increase. The last two are counter to what most published results show.

- Performance is worst when there is one outlying group along the regression line and another group at the same location in X space, but with significantly larger residual values.

- The number of clean observations classified as outliers (false alarms) decreases as the number of regressors increases. This false alarm rate increases as percentage of outliers and number of observations increase.

- The null case is a limitation to the methodology. When there are no outliers, then approximately 20% of the observations are identified as candidates for outliers.

2.4.11 Lee and Fung Forward Selection Algorithm

Lee and Fung (1997) propose a stepwise algorithm to detect multiple outliers in generalized linear models (GLIMs) and nonlinear regression based on a high breakdown robust estimator. They determine the clean data set from the studentized residual (GLIM raw residual over standard error) from a robust fit and sequentially add some of the initial outliers back to the clean set since too many outliers are identified. Outliers are added back by determining the upper 5% bound on the studentized residuals via Monte Carlo simulation. This procedure iterates until no observations exceed the 5% upper bound. There were no problems encountered in the selected examples, but further simulation is required to accurately assess finite sample performance.

2.4.12 Luceno Reweighted Least Deviances Algorithm

Luceno (1998) discusses using the weights from a reweighted least squares procedure to detect multiple outliers in the GLIM. The mean of the deviances (sum of squared deviance residuals) is replaced by a weighted mean of deviances. The weights are calculated with a Huber or redescending function. The parameter estimates come from minimization of the quantity $n^{-1} \sum_{i=1}^{n} w_i D_i(\mu; \phi; y)$. D_i is the squared deviance residual for the ith observation, μ is the mean ($\mathbf{X}\beta$ in normal theory), ϕ is the nuisance parameter (σ in normal theory models), and w_i is the weight from the influence function. If weights from Huber's function are used, then $w_i = 1.5/|D_i^{1/2}|$ if $|D_i^{1/2}| > 1.5$ otherwise $w_i = 1.0$. The procedure avoids estimating σ (or the appropriate nuisance parameter) by assuming detection of outliers is insensitive to σ within a certain range. Outliers are considered observations with unusually low values for the weights. Luceno suggests direct minimization of the objective function is computationally reasonable (when compared to LTS or LMS) and should be done on random subsets to avoid local minimums.

The procedure successfully detects outliers in several examples from McCullagh and Nelder (1989) and also identifies the outliers in the stackloss data set. The method appears to be effective at detecting leverage outliers. Performance apart from 4 examples is not reported.

2.5 Robust Regression

Either the residual or the observation's final weight from a robust estimator can be used to identify multiple outliers in regression. Robust regression accommodates outliers by judiciously downweighting them through the selection of model and input parameters. We also consider robust regression estimators beyond the purpose of outlier identification in this research. The literature on robust regression is vast and what follows is only a portion that is most directly applicable to this research.

2.5.1 Properties of Robust Regression Estimators

The three most important properties for robust regression estimators are breakdown, efficiency, and bounded-influence. The concept of breakdown is the primary motivation for using robust regression over OLS. The breakdown point is defined as the smallest fraction of anomalous data that can render the estimator useless. As displayed in Figure 1.1, a single outlying point can significantly change the OLS estimates of β ; the breakdown point is $1/n$, or 0% because n can be made arbitrarily large. Robust estimators can have breakdown points as high as 50%.

Another desirable property for robust estimators is efficiency. The efficiency is defined as the performance of the robust estimator relative to OLS under the assumption of no outliers; ε is NID $(0, \sigma^2 I)$. Recall that the OLS estimate will be the minimum variance estimate among all unbiased estimators. Typically, efficiency is expressed as the ratio of mean square errors.

The third desirable property is bounded-influence in X-space. This is the estimator's resistance to being "pulled" toward the extreme observations in X-space. Least squares is not bounded-influence and the more remote observations exert greater influence on the parameter estimates.

2.5.2 High-Breakdown Point Estimators

High-breakdown point (HBP) regression estimators have been developed to provide reliable estimates in the presence of a large percentage of outlying observations. These estimators can achieve up to a 50% breakdown point and are also know as resistant estimators. They are useful for outlier detection and initial estimators, but their low efficiency and unbounded influence deter from their use as stand-alone estimators.

Least Median of Squares (LMS) Estimators. Rousseeuw (1984) introduced the high-breakdown (as much as 50%) LMS estimators. LMS is obtained by minimizing the hth ordered squared residual where h is defined as the integer portions of $n/2 + (p+1)/2$. The objective function can be expressed as $\min_{\beta} median(e_i^2)$. LMS fits just over half the data and minimizes the residual for a single observation. The original proposal solved the objective function with random resampling; however, improved algorithms now exist (Burns, 1992 and Atkinson, 1994). The primary unattractive characteristic of LMS is an asymptotic efficiency of 0%. Although useful if severe contamination is suspected or when used in conjunction with other techniques, LMS has grown out of favor. Ryan (1997) argues against LMS based on an unstable algorithm (computationally intense and possibly different solutions using the same data) and small changes in the data result in

large changes in parameter estimates. He concludes LMS should not be used as a stand-alone estimator, an initial estimator or an outlier detection estimator.

Least Trimmed Sum of Squares (LTS) Estimators. Rousseeuw (1984, 1985) proposed the LTS high breakdown estimator as an efficient alternative to LMS. The LTS estimator is formed by minimizing the h out of n ordered squared residuals from smallest to largest. Rousseeuw and Leroy (1987) recommend $h = n(1-\alpha) + 1$ where α is the trimmed percentage. The objective function is $\min_{\beta} \sum_{i=1}^{h} \left(e_i^2\right)_{i:n}$ and it is solved with either random resampling (Rousseeuw and Leroy, 1987), a genetic algorithm (Burns, 1992) or forward search (Woodruff and Rocke, 1994). This estimator is attractive because α can be selected to prevent some of the poor results other 50% breakdown estimators show. LTS can be fairly efficient if the number of trimmed observations is close to the number of outliers because OLS is used to estimate parameters from the remaining h observations. The LTS estimator can become computationally intense as the number of observations increase.

S-Estimators. Rousseeuw and Yohai (1984) develop a high breakdown estimator (as much as 50%) that minimizes the dispersion of the residuals. The objective function is $\min_{\beta} s\left(e_1(\beta),...e_n(\beta)\right)$ where $e_i(\beta)$ is the ith residual for candidate β. This objective function is given by the solution to $(n-p)^{-1} \sum_{i}^{n} \rho\left(\frac{y_i - \mathbf{x}_i'\hat{\beta}}{s}\right) = K$ where K is a constant

$E_\Phi[\rho]$ with Φ defined as the standard normal. Rousseeuw and Yohai (1984) suggest a redescending influence function as $\rho(x) = \dfrac{x^2}{2} - \dfrac{x^4}{2c^2} + \dfrac{x^6}{6c^4}$ if $|x| \le c$ otherwise

$\rho(x) = \dfrac{c^2}{6}$. The parameter c is the tuning constant. Tradeoffs in breakdown and efficiency are possible based on choices for the tuning constant c and K. The usual choice is $c = 1.548$ and $K = 0.1995$ for 50% breakdown and about 28% asymptotic efficiency (Rousseeuw and Leroy, 1987).

The final scale estimate, s, is the standard deviation of the residuals from the fit that minimized the dispersion of the residuals. The scale estimate is an implicitly derived M-estimate of scale. Ruppert (1992) suggests an improved resampling algorithm and concludes that S-estimators perform marginally better than LMS and LTS.

2.5.3 *M*-Estimators and Multi-Stage Procedures

M-Estimators. M-estimators are maximum likelihood robust estimators proposed by Hampel (1973) that are nearly as efficient as OLS. Rather than minimize the sum of squared errors as the objective, the M-estimate minimizes a function ρ of the errors. The M-estimate objective function is $\min\limits_\beta \sum\limits_{i=1}^{n} \rho\left(\dfrac{e_i}{s}\right) = \min\limits_\beta \sum\limits_{i=1}^{n} \rho\left(\dfrac{y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{s}\right)$ where s is an estimate of scale often formed from a linear combination of the residuals. The system of normal equations to solve this minimization problem is found by taking partial

derivatives with respect to β and setting them equal to 0, yielding $\sum_{i=1}^{n} \psi\left(\dfrac{y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{s}\right)\mathbf{x}_i = \mathbf{0}$

where ψ is the derivative of ρ.

The choice of the ψ-function is based on the preference of how much weight to assign outliers (see e.g. Montgomery and Peck, 1992). A monotone ψ-function does not weight large outliers as much as least squares (e.g. a 10σ outlier would receive the same weight as a 3σ outlier). A redescending ψ-function increases the weight assigned to an outlier until a specified distance (e.g. 3σ) and then decreases the weight to 0 as the outlying distance gets larger.

Newton-Raphson and Iteratively Reweighted Least Squares (IRLS) are the two methods to solve the M estimates nonlinear normal equations. IRLS is the most widely used in practice and the only one considered for this research. IRLS expresses the normal equations as $\mathbf{X'WX}\hat{\boldsymbol{\beta}} = \mathbf{X'Wy}$ where $\mathbf{W}$ is an $n \times n$ diagonal matrix of weights

$w_i = \dfrac{\psi\left[\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_0\right)/s\right]}{\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_0\right)/s}$. The initial vector of parameter estimates, $\hat{\boldsymbol{\beta}}_0$, are typically obtained from OLS or a high-breakdown point estimator. IRLS updates these parameter estimates with $\hat{\boldsymbol{\beta}}_1 = \left(\mathbf{X'WX}\right)^{-1}\mathbf{X'Wy}$. The procedure continues until some convergence criterion is satisfied. The estimate of scale may be updated after the initial estimate.

Generalized M-Estimators. The Generalized M-estimators (*GM*), proposed by Mallows (1975) and improved by Krasker and Welsch (1982), were developed to overcome the limitations of M-estimators for high-leverage observations. The *GM-*

estimator bounds the influence in X-space by weighting the M-estimate system of normal equations by a measure of leverage. The GM system of normal equations is

$$\sum_{i=1}^{n} \pi_i \psi\left(\frac{y_i - \mathbf{x}'_i\hat{\beta}}{s\pi_i}\right)\mathbf{x}_i = \mathbf{0}$$ where π_i is a measure of remoteness in X-space. When the π-

weights are located both inside and outside the argument of the ψ-function, the GM objective function is Schweppe (Handsin et al., 1975). If the π-weights are not inside the argument, the GM objective function is Mallows (Mallows, 1975). In practice, the distinction between the two objective functions is that Mallows will downweight high-leverage points independently of the residual value while Schweppe will not downweight if the response value is in line with the regression plane. Thus, Mallows does not fully incorporate "good outliers" in the parameter estimates. There are several approaches to forming the π-weights that use some form of the leverage measures discussed in Section 2.5.4.

A numerical optimization scheme is required to solve the GM system of nonlinear normal equations. The two most common approaches are also Newton's method and IRLS as in M-estimation. The initial parameter estimates are most often from one of the HBP estimators in Section 2.5.2. The final parameter estimates can come from a fully iterated solution (GM-estimator) or only a single iteration (compound estimator). The single iteration method preserves the breakdown of the initial estimator (Simpson, Ruppert, and Carroll, 1992).

MM-Estimators. Yohai (1987) and Yohai et al. (1991) introduce *MM* estimators that achieve the high-efficiency of *M*-estimators and are also high-breakdown. The first stage of the three stage procedure calculates an *S*-estimate with influence function

$$\rho(x) = 3\left(x/c\right)^2 - 3\left(x/c\right)^4 + \left(x/c\right)^6 \text{ if } |x| \le c; \text{ otherwise } \rho(x) = 1.$$ The value of the tuning constant, c, is selected as 1.548. The second stage calculates the *MM* parameters that

provide the minimum value of $\sum_{i=1}^{n} \rho\left(\frac{\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_{MM}\right)}{\hat{\sigma}_0}\right)$ where $\rho(x)$ is the influence function

used in the first stage with tuning constant 4.687 and $\hat{\sigma}_0$ is the estimate of scale from the first step (standard deviation of the residuals). The final step computes the *MM* estimate

of scale as the solution to $(n-p)^{-1} \sum_{i=1}^{n} \rho\left(\frac{\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}\right)}{s}\right) = 0.5.$

The *MM* estimator generally performs well except in areas with high-leverage (Simpson and Montgomery, 1998b). *S-Plus* version 4.5 has included the *MM* estimator with the Yohai et al. (1991) test for bias on the robust regression menu.

2.5.4 Leverage Measures in Robust Regression

One objective of this research is to improve multi-staged *GM* and compound estimators. Another factor to improve, other than the HBP initial estimator, is the π-weights that measure the remoteness in X-space. Some other measures of leverage beyond the hat diagonals and the Mahalanobis Distance used in robust regression are the *M*-estimates of covariance, the minimum volume ellipsoid (MVE), and the minimum covariance determinant (MCD).

M-Estimates of Covariance. Hampel (1973) first suggested *M*-estimates of covariance, but the basic paper on these estimators is attributed to Maronna (1976). Maronna addressed the problems of existence, uniqueness, asymptotic distribution and breakdown point for these estimators. We are interested in the distances in X-space for each observation defined by $z = \hat{A}(x - \hat{t})$ where $\hat{A}$ is an estimate of the $p \times p$ multivariate scatter matrix and $\hat{t}$ the multivariate location vector. Note that $(\hat{A}'\hat{A})^{-1}$ is the estimate of the covariance matrix of **X**. From Huber (1981), the maximum likelihood estimate of **A** and **t** is determined by solving the simultaneous equations

$$\text{ave}\{w(|z|)z\} = 0$$

$$\text{ave}(\{u|z|)zz^T - v(|z|)I_p\} = 0$$

where u, v and w are arbitrary weight functions and ave$\{\cdot\}$ is the average taken over the sample. We solve these equations using the Newton algorithm and Huber weight functions with the associated constants and correction factors as defined in the ROBETH library (Marazzi, 1993).

The following steps summarize the ROBETH library implementation of the *M*-estimates of covariance procedure to compute robust distances for the $n \times p$ matrix **X** with elements x_{ij} for $i = 1$ to n and $j = 1$ to p.

1. Find initial estimates of **A**, and **t**. $\hat{t}_j = \underset{i}{med}\{x_{ij}\}$ and $\hat{A}$ is a diagonal matrix with diagonal elements $\hat{a}_{jj} = 1/med\{|x_{ij} - \underset{i}{med}\{x_{ij}\}|\}/0.6745$.

2. Find the constant parameters (a, b, c, d) of the arbitrary weight functions $u(z)$ (Huber's weight function), $v(z)$, and $w(z)$ by first specifying the expected proportion of outliers in the sample, ε.

$$u(z) = \begin{array}{ll} a^2/z^2 & \text{for } z^2 < a^2 \\ 1 & \text{for } a^2 \le z^2 \le b^2 \\ b^2/z^2 & \text{for } b^2 < z^2 \end{array}$$

$a^2 = \max(p - \kappa)$ and $b^2 = p + \kappa$. The value for κ comes from regula-falsi solution to

$$\frac{1}{1-\varepsilon} = k_p \left(\frac{1}{\sqrt{2\pi}}\right)^p \left\{ e^{-a^2/2} \int_0^a \left(\frac{a}{r}\right)^{a^2} r^{p-1}dr + \int_a^b e^{-r^2/2} r^{p-1}dr + \int_b^\infty \left(\frac{b}{r}\right)^{b^2} r^{p-1}dr \right\}$$

where k_p denotes the surface of the unit sphere in dimension p; $k_p = 2\pi^{p/2}/\Gamma(p/2)$.

$$v(z) = d \qquad \text{for all } z$$
$$d = 1/p\left\{a^2 \chi_p^2(a^2) + b^2\left(1 - \chi_p^2(b^2)\right)\right\} + \left\{\chi_{p+2}^2(b^2) - \chi_{p+2}^2(a^2)\right\}$$

$$w(z) = \begin{array}{ll} 1 & \text{for } z < c \\ c/z & \text{for } z \ge c \end{array}$$

A Newton procedure solves for c in $e^{-c^2/2}/\sqrt{2\pi} + c(\Phi(c) - (1 - \varepsilon/2)/(1 - \varepsilon)) = 0$

3. Calculate
$$z_i = \hat{A}(x_i - \hat{t}) \qquad \text{for } i = 1 \text{ to } n$$
$$r_j = \sum_i w(|z_i|)(x_{ij} - t_j) \qquad \text{for } j = 1 \text{ to } p$$
$$s_2 = \sum_i \left\{w(|z_i|) + w'(|z_i|)|z_i|/p\right\}$$

4. Compute a lower triangular matrix of improvements $S = (s_{jk})$

$$s_{jj} = \frac{p}{2(\tilde{a} + b)}\left(a_{jj} - (\tilde{b} - \tilde{c})\tilde{e} - \tilde{d}\right) \qquad j = 1 \text{ to } p$$

$$s_{jk} = \frac{p}{(\tilde{a} + \tilde{b})}a_{jk} \qquad j > k$$

$$s_{jk} = 0 \qquad j < k$$

where $\tilde{a} = n^{-1}\sum_i u(|z_i|)|z_i|^2$; $\tilde{b} = (n(p+2)^{-1})\sum_i u'(|z_i|)|z_i|^3$;

$\tilde{c} = n^{-1}\sum_i v'(|z_i|)|z_i|$; $\tilde{d} = n^{-1}\sum_i v(|z_i|)$; and $\tilde{e} = \dfrac{\tilde{d}p - \tilde{a}}{2(\tilde{a} + \tilde{b}) + p(\tilde{b} - \tilde{c})}$

5. Update the location estimate, $\hat{t}_j = \hat{t}_j + h_j$ where $h_j = r_j/s_2$ from step 3.

6. Update the scatter matrix, $\hat{A} = (I - \gamma S)\hat{A}_0$ where $\hat{A}_0$ is the initial (or current) estimate of the scatter matrix and γ is the step length based on the maximum value of s_{jj}.

7. Check for convergence on the location and scatter matrix estimates and go to step 3 if improvement is still possible based on the specified tolerance values. Otherwise

calculate the distances from the M-estimates of covariance with the current location and scatter estimates.

Minimum Volume Ellipsoid (MVE). Rousseeuw (1985) proposes the MVE as a high-breakdown estimate for the mean and covariance matrix. The MVE is the smallest ellipsoid covering just over half of the data. The robust estimates of the mean and covariance matrix come from the classical calculation of these quantities only using the subset of observations that are contained in the MVE. The original algorithm uses random resampling to find the subset of observations that is covered with the smallest volume ellipsoid. The algorithm is:

1. Form a random sample of size $q = p + 1$ from the n observations.

2. Compute the classical mean and covariance matrix for this sample of size q.

3. Compute the Mahalanobis distance for all $\dot{n}$ observations with the estimators in (2).

4. Increase the sample to size $n/2 + 1$ by adding the $n/2 + 1 - q$ observations with the least Mahalanobis distance from (3) and compute the mean and covariance matrix.

5. Compute the product of the median Mahalanobis distance and the covariance matrix from (4).

6. The determinant of the quantity in (5) is proportional to the volume of the ellipsoid covering these observations.

7. Iterate steps 1 through 6 for the specified number of random samples to evaluate.

8. Compute the mean vector and covariance matrix for the sample that yields the minimum value for the quantity in 6.

9. Correct the covariance matrix for small sample sizes (Rousseeuw and van Zomeren, 1991) and consistency at multivariate normal distributions with the quantity $\dfrac{(1+15/(n-p))^2}{\chi^2_{p-1,0.50}}$.

Hawkins (1993) improved the algorithm using steepest descent with random restarts rather than the random sampling method. Woodruff and Rocke (1993) propose a heuristic search optimization procedure. Currently *S-Plus 4.5* uses genetic algorithms (Burns, 1992).

Unfortunately, the MVE is inefficient with asymptotic efficiency of 0 (Davies, 1992). The implementation in *S-Plus* goes an additional step to increase efficiency. All remaining observations apart from those in the MVE are added back to compute the final estimate of the mean and covariance matrix if their Mahalanobis distances (calculated with the original MVE estimates) are less than a cutoff value from the chi-square distribution. This significantly increases the efficiency of the estimates. An alternative estimator is the Minimum Covariance Determinant (MCD) that was also introduced by Rousseeuw (1985). The MVE is an $n^{-1/2}$ estimator and the MCD is $n^{-1/3}$ (Butler et al., 1993).

Minimum Covariance Determinant (MCD). The MCD searches for the sample of size $q < n$ that has the minimum value, among all samples evaluated, of the determinant of its covariance matrix. The estimator is most often a 50% breakdown estimator so q is set to the integer part of $(n + p + 1)/2$. The algorithm has evolved from random resampling much like the MVE. The improvements proposed by Hawkins (1994), Woodruff and Rocke (1994) and the genetic algorithms (Burns, 1992) parallel those of the MVE. Butler et al. (1993) prove that the MCD has much better statistical properties, notably efficiency, than the MVE. Several other authors recommend the MCD over the MVE (Simpson and Chang, 1997, simulations in Rocke and Woodruff, 1994); however, Rocke and Woodruff (1997) do not recommend either as stand-alone procedures because of computational complexity in high-dimension. It is not known how the genetic algorithms perform as stand-alone procedures. Rocke and Woodruff (1997) recommend their hybrid procedure from 1996.

Rocke and Woodruff Hybrid Procedure. Rocke and Woodruff (1996) propose a complex algorithm to detect outliers from multivariate normal samples large in both dimension and the number of observations. Their procedure combines several results from the literature to form a hybrid robust estimator of location and scale with attractive properties. This estimate is then used to compute the robust distance only for each observation and does not consider regression data and residuals. The estimator has up to a 40% breakdown point compared to the usual breakdown of $1/(1+p)$ for robust estimators. Additionally, the estimator is affine equivariant so linear transformations on

the data will not affect the performance. This measure of leverage has not been used in a *GM* or compound estimator.

Rocke and Woodruff use a two-phase approach. The output of the first phase is an estimate of multivariate location and shape. The first step is to equally partition the data into cells to minimize computational burden. Within each cell, the observations from the minimum covariance determinant (MCD) using Hawkins (1993) steepest descent algorithm with random restarts are the starting point for the sequential point addition algorithm from Hadi (1992). This result is then used as a starting point for the translated bi-weight *M*-estimation of the mean and covariance matrices. Rocke and Woodruff (1993) use their previously published simulation results to justify using the constrained *M*-estimator over the bi-weight *S*-estimator. The robust covariance and location matrices are found by using the estimators from the cell with the minimum determinant of the sample covariance matrix.

The second phase runs a simulation to determine the appropriate cutoff value to classify observations as outliers based on n observations in p dimensions using clean multivariate normal data in the Phase I algorithm. New location and shape matrices are formed by those observations below the simulated cutoff value. The robust distance is calculated using these new location and shape matrices and compared to a $\chi^2_{p,1-\alpha}$ critical value to classify the observation as outlying or not.

Their results show no problems with swamping when no outliers are present based on simulations with 10 - 40 variables and samples sizes from 50-3200. The proposed hybrid estimator significantly outperformed Rousseeuw's (1985) random search

over elemental subsets and marginally outperformed the forward search of Hadi (1992).

Their algorithm worked well on the smaller published "challenging" sets. Other

significant results for their procedure include:

- Identification of outliers is easier if the outliers lie in more than one cluster.

- In higher dimensions, outlier detection is more difficult, more data is required, and the breakdown is lower.

- Increasing sample size increases the probability of correctly identifying outliers.

- For reasonable computation time, breakdown is roughly 30-40% in dimension 10, 25-35% in dimension 20, and 20-25% in dimension 40.

2.6 Variable Selection Procedures

An important aspect of building a regression model is to decide which regressor

variables should be included in the model. The β vector is partitioned into an active

variable set, β_1 of $p - q$ parameters and inactive set β_2 of q parameters to test the

hypothesis that $\qquad$ H_O: $\beta_2 = 0$

$\qquad\qquad\qquad$ H_A: $\beta_2 \neq 0$.

Failure to reject the null hypothesis suggests there is no evidence that any of the regressor

variables in set β_2 have any affect on the response value.

The goal of a variable selection procedure is to have the significant regressor

variables included in set β_1 with high probability, while simultaneously achieving a high

probability that the insignificant variables are contained in set β_2. The regression model

building strategy is an iterative process that involves selection of an active subset of the p

parameters followed by model diagnostics to assess the fit. The objective is to find the

best subset of the p parameters to include in the model that leads to good prediction

capability yet minimizes the variance of prediction. The former objective would suggest

including all p variables while the latter suggests using as small of a subset as possible

because the variance of prediction always increases as regressor variables are added to a

model. Models with fewer variables are also preferred for simplicity and ease of data

collection.

2.6.1 Variable Selection in Regression

There are numerous variable selection methods available to the analyst. The

simplest is to retain only the variables whose ratio of coefficient to standard error is

significant. This t-test approach is not reliable as dimension increases and particularly

when dependencies between regressor variables exist. A common alternative is the class

of computer-intensive variable selection methods (e.g. forward, backward, stepwise, and

best subsets regression). The selection criteria are often based on F-tests (F-to-enter and

F-to-leave) or Mallows's (1973) C_p criterion; $C_p = \hat{\sigma}^{-2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 - n + 2p$ where $\hat{y}_i$ is

the predicted value and $\hat{\sigma}^2$ is typically the MSE from the full model. Unfortunately,

Miller (1990) demonstrates that the F tests and Mallow's C_p criterion are poor for model

selection as are the R^2 and adjusted R^2 measures. Breiman (1995) states that the preferred measure of performance for variable selection in regression is prediction error.

Resampling methods are currently recommended to calculate a measure of prediction error for variable selection. The two most common resampling methods are cross-validation and bootstrapping. Cross-validation procedures partition the data into two disjoint sets. The model is fit with one set (the training set) and subsequently used to predict the responses for the observations in the second set (assessment set). Bootstrap procedures form many samples from the original data by resampling with replacement. Details of the methods and their application to the variable selection problem in regression are outlined below.

2.6.2 Cross-Validation Procedures

An intuitively appealing method to calculate a predicted response value is to use the parameter estimates from the fit obtained by omitting the observation. This predicted response value is denoted by $\hat{y}_{(i)}$ and $\hat{\Delta}_{CV,1} = n^{-1}\sum_{i=1}^{n}(y_i - \hat{y}_{(i)})^2$ is known as the leave-one-out cross-validation estimate of average prediction error for a model. Apart from the n^{-1} term, this quantity is the PRESS statistic in least squares. For OLS, the PRESS statistic is calculated as $\sum_{i=1}^{n}\left(\dfrac{e_i}{1-h_{ii}}\right)^2$ where $h_{ii} = \mathbf{x}_i(\mathbf{X'X})^{-1}\mathbf{x}_i'$. Note that PRESS does not require n separate fits while other regression estimators (e.g. robust) do require all n fits for the leave-one-out cross-validation estimate of prediction error. Shao (1993) proves with

asymptotic results and simulations that the model with the minimum PRESS statistic or leave-one-out cross-validation estimate of prediction error is often overfit. He recommends using K-fold cross-validation that leaves a subset of observations out.

Quenouille (1949) explored the idea of leaving two observations out of the training set and Stone (1974) extended the method to more than two. In K-fold cross-validation, the training set omits approximately n/K observations from the training set rather than a single observation like PRESS. To predict the values for the kth assessment set, $S_{k,a}$, all observations apart from those in set k are the training set, $S_{k,t}$, and are used to estimate the model parameters. The K-fold cross-validation average prediction error is

$$\hat{\Delta}_{CV,K} = n^{-1} \sum_{i=1}^{n} \left(y_i - \hat{y}_{(k,i)}\right)^2 \text{ where } \hat{y}_{(k,i)} \text{ is the predicted response for observation } i$$

belonging in assessment set $S_{k,a}$.

One approach to the K-fold cross-validation estimate of prediction error is to randomly select the n/K observations to form the assessment set. This process is repeated numerous times and the prediction errors are averaged. Breiman et al. (1984) propose a less computationally intense scheme that randomly partitions the data into K different disjoint sets. Davison and Hinkley (1997) recommend $K = min\ (n^{1/2}, 10)$ in practice. This procedure decreases the variance of prediction error over that of the leave-one-out cross-validation estimate but at the expense of increased bias. Surprisingly, Shao (1993) demonstrates that the smaller the training set (larger value of K), the better the K-fold estimate is for model selection.

To reduce the bias, Burman (1990) recommends the adjusted K-fold cross-validation estimate of prediction error as

$$\hat{\Delta}_{ACV,K} = \hat{\Delta}_{CV,K} + \hat{\Delta}_{App} - \sum_{k=1}^{K} p_k \left(n^{-1} \sum_{i=1}^{n} (y_i - \hat{y}_{(k,i)})^2 \right)$$ where p_k is the ratio of observations

in assessment set k to the total n and $\hat{y}_{(k,i)}$ is the predicted response for the ith observation

from the fit with training set $S_{t,k}$. The Breiman and Spector (1992) simulations

demonstrate that the performance of the adjusted cross-validation prediction error

estimate is slightly worse than the standard K-fold cross-validation prediction error for

least squares variable selection. Shao (1993) shows that both the leave-one-out and K-fold cross-validation procedures have a negligible probability of selecting an

underspecified model. The challenge is avoiding an overfit model.

2.6.3 Bootstrap Procedures

Bootstrap estimators in regression have received considerable attention in the

literature since their introduction by Efron (1979). Wu (1986) provides the theoretical

results for bootstrap methods applied to regression. Hall (1989) proves that inference in

regression, such as confidence intervals, based on the bootstrap estimate are more

accurate than standard inference procedures even if the error is Gaussian.

The fundamental element of a bootstrap procedure is the bootstrap sample. For

bootstrapping pairs in regression (Efron, 1982), the sample is formed by randomly

sampling with replacement n times both a response and its associated vector of regressor

variable values from the original sample. The bootstrap sample may contain an

observation from the original sample once, multiple times or not at all. In fact, the

probability that an observation is included in a bootstrap sample of size n is $1 - e^{-1} =$

0.632 (Efron and Tibshirani, 1997). A regression model is then fit to the bootstrap

sample to obtain the bootstrap parameter estimates $\hat{\beta}^*$. A large number of bootstrap

samples (B $\geq$ 100) are constructed from the original sample for model inference.

For the variable selection problem, the estimate of average prediction error for the

b^{th} bootstrap sample is $\hat{\Delta}_b = n^{-1}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\hat{\beta}^*_b)^2$ where $y_i, \mathbf{x}_i$ are from the original sample.

Efron (1983) provides the unbiased estimator of prediction error as

$$\hat{\Delta}_{b,unbiased} = n^{-1}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\hat{\beta})^2 + n^{-1}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\hat{\beta}^*_b)^2 - n^{-1}\sum_{i=1}^{n}(y_i - \mathbf{x}^*_i\hat{\beta}^*_b)^2$$ where $\mathbf{x}^*_i$ is the

vector of regressor values for the ith observation in the bth bootstrap sample. The overall

bootstrap estimate of average prediction error is simply $\hat{\Delta}_{BS} = B^{-1}\sum_{b=1}^{B}\hat{\Delta}_{b,unbiased}$. Shao

(1996) shows that selecting the model with the minimum $\hat{\Delta}_{BS}$ is inconsistent.

Inconsistency implies that the probability the true model has the minimum bootstrap

average prediction error does not equal 1.0 as n approaches infinity. Shao corrects this

inconsistency for bootstrapping pairs by using substantially fewer than n observations to

construct the bootstrap samples. This procedure does not use the bias-corrected estimate

of prediction error. Breiman (1996), motivated by increasing the 0.632 probability that

an observation is selected in a bootstrap sample, notes that using bootstrap samples of

size $2n$ has little effect on the results for least squares variable selection.

2.6.4 Other Modifications to Resampling Methods for Variable Selection

Breiman and Spector (1992) explore the use of cost admissibility (penalty for adding variables) with bootstrap and cross-validation prediction error for variable selection. Their empirical results indicate that this modification is only slightly beneficial to the variable selection process. This is an important result because most resampling estimates of prediction error do not account for the number of variables in the model.

Breiman (1992) recommends the little bootstrap estimate of prediction error for variable selection in linear models. The prediction error for a k variable model using this approach is $\hat{\Delta}_{App}(k) + 2B_t(k)$. The little bootstrap error, $B_t(k)$, is the resubstitution error from the model selected using $\mathbf{y}^* = \mathbf{y} + \tilde{\epsilon}$ where $\tilde{\epsilon}$ is a vector of variates from NID (0, $t^2\sigma^2$) with $0.6 < t < 0.8$. The MS_E for the full model is used as an estimate of σ^2. Breiman shows that the little bootstrap is unbiased and superior to C_p, F-to-enter, and F-to-leave for variable selection for fixed designs.

Breiman (1996) suggests *bagging* (bootstrap aggregating) regressor variables. For each of the B samples formed by bootstrapping pairs, perform a forward selection to obtain a 1 variable model, 2 variable model, ... k variable model. The $n \times k$ matrices of predicted values from these k models are averaged across the B bootstrap samples. The model with the lowest average prediction error is selected. Limited simulation results indicate that this procedure performs better than standard forward selection. It is unclear how to proceed if the same variables are not consistently selected in the B samples for a given dimension.

Davison and Hinkley (1997) describe a hybrid estimate of bootstrap prediction error for variable selection adapted from Efron and Tibshirani (1997). The hybrid estimate of prediction error weights the apparent error and the bootstrap cross-validation error calculated from the predicted values of those observations not included in the bootstrap sample. The authors' empirical evidence suggests this procedure is superior, although no results are published.

2.6.5 Variable Selection with Robust Regression Estimators

Although numerous estimators have been proposed in the last 25 years, there are significantly fewer results in the literature that explore variable selection procedures in the robust regression model. Most robust regression variable selection methods are based on robust versions of the general linear test that use the asymptotic covariance matrix (Hampel et. al, 1986). Markatou and He (1994) and Hertier and Ronchetti (1994) extend the Wald (similar to t-tests) and drop-in-dispersion tests (similar to F-tests) to GM and compound estimators. Field (1997) and Field and Welsh (1998) propose saddlepoint approximations of tail area probabilities for robust regression hypothesis testing as improvements to the asymptotic approach. The results are mixed and they recommend further testing in finite samples. Ronchetti and Staudte (1994) propose a robust version of Mallows's C_p. This method multiplies the squared residuals by the final weights from a robust fit to compute the residual sum of squares. Two additional quantities are also added to the residual sum of squares that are a function of the number of parameters and

the selected robust estimator. The robust C_p appears to work satisfactorily for their three examples, but no simulation results are reported.

The Wald test is currently preferred (Hertier, 1997) because of its asymptotic chi-square distribution and the relative ease to calculate the asymptotic covariance matrix. Wilcox (1997) experiments (results not reported) with the Wald test using the M-estimator and the Coakley and Hettmansperger (1993) compound estimator. He found for both estimators, even with normal and homoscedastic error terms and $n = 100$, poor control over Type I error. All authors conclude that it is important to do further testing and evaluation to understand the strengths and weaknesses of the methods in finite samples.

A common use of resampling methods in robust regression is construction of confidence intervals and prediction intervals with the bootstrap (Efron and Tibshirani, 1993, Davison and Hinkley, 1997, Wilcox, 1994, 1996a, 1996b, 1997). Mammen (1993) shows the consistency of the bootstrap for linear tests with the M estimator.

Wilcox (1997, 1998) presents an interesting approach to the variable selection problem in robust regression using a bootstrap resampling scheme. He uses a percentile bootstrap approach to find critical values for the joint confidence region of the Mahalanobis distance for the model parameters. The steps of the algorithm are:

1. Obtain B bootstrap estimates of β by bootstrapping pairs.
2. Estimate the covariance matrix V using all B bootstrap estimates of β .
3. Find the Mahalanobis distance of $(\beta^* - \hat{\beta})$ using V^{-1} for each bootstrap sample where β^* is the bootstrap estimate of the model parameters and $\hat{\beta}$ is the vector of parameter estimates from the original data.
4. Sort the Mahalanobis distances and call the $(1-\alpha)B$ ordered distance the critical value.

5. Find the test statistic by the Mahalanobis distance using V^{-1} of $(\hat{\beta} - c)$ where c is a vector of constants often selected as 0 to test for significance.

Wilcox (1998) states there is room for improvement with this method because the probability of a Type I error can be substantially less than nominal levels in many circumstances. He states that this approach does not work well with least squares; correction factors through simulation are required to achieve the correct coverage probabilities.

Davison and Hinkley (1997) provide a brief discussion of resampling methods in robust regression. Their guidance on resampling methods for variable selection in robust regression focuses on two main points: 1) remove gross outliers from analysis because too many outliers could appear in the resampled data leading to inefficiency and breakdown and 2) most of the prediction error methods for least squares *should* apply to robust regression. They recommend that gross outliers be removed by large residuals from an LTS fit.

2.7 Literature Review Summary

This chapter has reviewed the relevant published results to the research objectives. Clearly, there are numerous options available for the multiple outlier detection problem with few comparable results available between methods. A comprehensive performance study is missing. Also, several options exist for the selection of components in multi-staged *GM* and compound estimators. A critical evaluation of these components could lead to improved performance. Lastly, the variable

selection problem has not been fully explored for multi-staged *GM* and compound

estimators. The usefulness of variable selection resampling methods has not been

thoroughly investigated.

Chapter 3

A Comparative Analysis of Multiple Outlier Detection Procedures

3.1 Introduction

There has been considerable interest in recent years in the detection and accommodation of multiple outliers in statistical modeling. This chapter uses Monte Carlo simulation to evaluate numerous recently published outlier techniques in the linear regression model. Kianifard and Swallow (1990) report a similar smaller study using a few earlier techniques. Other comparative analyses typically appear in journal articles where the authors propose a new methodology; however, these studies are often limited in scope and breadth of techniques. Our approach tests the latest and most respected multiple outlier detection procedures across a number of realistic and challenging regression scenarios.

In general, Barnett and Lewis (1994) define outliers as observations that appear inconsistent with the remainder of the data set. For this paper, we wish to identify outliers in *linear regression* modeling. Specifically, we are concerned with observations that differ from the regression surface defined by the bulk of the data. It is important to identify these types of outliers in regression modeling because the observations, when undetected, can lead to erroneous parameter estimates and inferences. Additionally, these outliers may be of interest themselves to provide insight into process behavior at certain operating conditions.

If only a single or few outliers exist, many standard least squares regression diagnostic quantities and plots will reliably identify these observations. However, these diagnostics have been shown to fail in the presence of multiple outliers; particularly if the observations are clustered in an outlying cloud. The measures may either fail to identify the outliers (masking), identify the clean observations as outliers (swamping), or could both mask and swamp observations. To overcome the limitations of the standard least squares diagnostics, numerous multiple outlier detection techniques have been proposed to identify the outlying subset of observations.

The outlying observations can be remote in the levels of the regressor or explanatory variables (exterior X-space observations). These are considered high-leverage points because they are influential and pull the regression surface toward them. We refer to cases that are not unusual in X-space as interior X-space observations. Observations can also be outlying in the response variable (Y-space) because of distant values from the responses of the clean cases. Further classification of outliers is possible with respect to the regression model. If the observations do not conform to the regression surface defined by the bulk of the data, then these cases are known as regression or residual outliers. We are concerned with two main outlier configurations likely to be encountered in practice: 1) observations that are interior X-space regression outliers and 2) observations that are exterior X-space regression outliers. We consider testing these scenarios when the response variable for the outliers is a Y-space outlier and when it is not. A third important outlier scenario occurs when the observations are remote in X-space but the response

values conform to the regression surface. We limit the scope of this chapter by not including these high-leverage "good outliers" in the study.

Section 3.2 briefly describes the multiple outlier detection procedures used in this comparative study. Detailed summaries of many of these and other multiple outlier detection procedures can be found in Chapter 2, Hadi and Simonoff (1993), Barnett and Lewis (1994), and Sebert (1997). Section 3.3 describes the Monte Carlo simulation scenarios, factors, factor settings and the measures of performance; Section 3.4 provides the simulation results and analysis; and Section 3.5 summarizes the results for each procedure and provides recommendations.

3.2 Multiple Outlier Detection Procedures

The multiple outlier detection methods for linear regression selected in this study are either those most recently published or those most frequently cited in the literature. We do not consider many of the previously-published methods that have been tested and proven to be either ineffective or too restrictive in assumptions (e.g., specifying the exact number of outliers). We do not consider (but do advocate) the subjective evaluation of the data from various multivariate plots to identify the outliers as suggested by Atkinson and Riani (1997) and Cook (1998), among others.

It is convenient to consider two broad classes of multiple outlier detection procedures as defined by Hadi and Simonoff (1993): direct methods and indirect methods. The direct methods use algorithms to isolate outliers and the indirect methods use the results from robust regression estimators. The description of both the direct and indirect

procedures below considers the standard linear model $y = x\beta + \varepsilon$ where y is the observed response vector of dimension n, the number of observations; X is the observed $n \times p$ matrix of regressor variables with intercept; and ε is the column vector of n random errors assumed to have mean 0 and covariance matrix $\sigma^2 I$.

3.2.1 Direct Procedures

Many of the direct procedures in the literature are based on either sequential deletion (backward search) of outlying observations or sequential addition (forward search) of clean observations. In a backward search, the entire set of observations is initially considered and the outliers are sequentially removed by a criterion such as the largest absolute value of some transformed residual. The forward search works similarly. A small subset of the data is selected as the initial clean basis and clean observations are sequentially added to this basis. Methods using forward search generally outperform backward search methods (Simonoff, 1991, Atkinson and Riani, 1997). We consider the forward search procedures from Hadi and Simonoff (1993, 1997) and Swallow and Kianifard (1996). We also consider the direct procedure based on the eigenstructure of the influence matrix from Pena and Yohai (1995) and the clustering algorithm from Sebert et al. (1998). The general steps of these algorithms and specific issues related to this research are outlined below. For most of these procedures, the authors provide alternative algorithms and parameter settings. Our philosophy is to choose the best performing options determined from our pilot studies, the authors' published results or both.

The Hadi and Simonoff (1993) forward search algorithm. This procedure initially determines a clean basis of $p + 1$ observations from the smallest absolute value of the adjusted residual from a least squares fit, $a_i = e_i / \sqrt{1 - h_{ii}}$. This basis is iteratively increased to the initial clean subset of size $v = (n + p + 1)/2$ by using the lowest values in magnitude of least squares scaled residuals. Next, the absolute values of the studentized residual (if the observation is in the current basis, M) or the scaled prediction error (if the observation is not in M) are ordered and the lowest $v + 1$ cases become the new basis M. The procedure continues to add observations until the $(s + 1)^{st}$ ordered residual measure exceeds $t_{(\alpha/2(s+1),s-k)}$ where s is the number of observations in the current subset. Observations $s + 1$ to n are the outliers. Hadi and Simonoff (1997) improve this algorithm by using the robust distance measures from the Hadi (1992, 1994) forward selection algorithm to determine the initial clean subset of size v observations.

The Swallow and Kianifard (1996) recursive residual forward search algorithm. Swallow and Kianifard suggest recursive residuals standardized by a robust estimate of scale as the test statistic to classify multiple outliers. The algorithm first orders the magnitudes of the studentized residual values from a least squares fit to form the basis of p clean observations. Recursive residuals, w_j, are scaled by the median absolute deviation from the median (MAD) estimate of scale $\hat{\sigma}$. The w_j are defined as

$$ w_j = \frac{y_j - \mathbf{x}' \hat{\boldsymbol{\beta}}_{j-1}}{(1 + \mathbf{x}'_j \, (\mathbf{X}'_{j-1} \mathbf{X}_{j-1})^{-1} \mathbf{x}_j)^{1/2}}, j = p+1,...,n . $$

The MAD is median $\{|e_i - \text{median } \{e_i\}|\}$ where e_i is the OLS residual, not the studentized residual.

The test statistic $\left|w_j / \hat{\sigma}\right|$ for each observation is compared to a cutoff value to identify the outliers. A correction factor for the MAD estimate of scale and the cutoff value come from simulation under the null hypothesis of no outliers.

The Pena and Yohai (1995) influence matrix algorithm. This procedure searches for breakpoints in the ordered components within the eigenvectors from the influence matrix, $\mathbf{M} = \mathbf{EDHDE}/ps^2$ where $\mathbf{E}$ is the diagonal matrix of least squares residuals, $\mathbf{D}$ is the diagonal matrix with elements $(1 - h_{ii})^{-1}$, $\mathbf{H}$, the hat matrix, $= \mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'}$, and s^2 is the usual mean square error estimate of the variance. If the ratio of components exceeds 2.5, then consider all ordered observations after (or before if the components are negative) this breakpoint as the candidate outliers.

The Sebert, et al. (1998) clustering algorithm. This approach clusters the standardized predicted and standardized residual values from a least squares fit. The crux of the algorithm is finding the single largest cluster, or the bulk of the data to classify as the inliers. Mojena's stopping rule forms the final clusters (single linkage, Euclidean distance) by splitting a cluster tree at the average of the $n - 1$ tree cluster heights (a measure of cluster separation) plus 1.25 times the standard deviation of the tree cluster heights.

3.2.2 Indirect Procedures from Robust Regression Estimators

Robust regression techniques accommodate outliers by downweighting or ignoring the unusual observations to ensure they are not too influential on the regression parameter estimates. It is possible to detect aberrant observations from either the final weights assigned to the observations or by the magnitude of the residuals. Our research has shown the residuals provide the most reliable signal to detect multiple outliers. The cutoff values to declare an observation an outlier from the residual value must be computed by Monte Carlo simulation because the distribution of robust regression residuals is not known. We generate 1000 clean data sets from the specified distribution with k regressor variables and n observations under the null hypothesis of no outliers. The cutoff value is the average of the two appropriate percentiles (e.g., the 2.5[th] and 97.5[th]) of the $1000 * n$ residuals. All robust regression estimators in this research have nearly symmetric distributions for the residuals of clean observations.

The multiple outlier detection capability of several common robust regression estimators is tested in several scenarios. The common robust estimators are Least Median of Squares (LMS), Least Trimmed (sum of) Squares (LTS), and M-estimators. These three estimators are available using the internal functions of *S-Plus 4.5*. We also consider the *MM* estimator from Yohai (1987) and its implementation through the ROBETH *S-Plus* library (Marizzi, 1993). We use the code from Wilcox (1997) for the standard bounded influence generalized M-estimator and the compound estimator from Coakley and Hettmansperger (1993). Also tested is the Simpson and Montgomery (1998) compound estimator.

LMS estimator. Rousseeuw (1984) introduced the high-breakdown (as much as 50%) LMS estimators. LMS is obtained by minimizing the hth ordered squared residual where h is defined as the integer portions of $n/2 + (p+1)/2$. Note h is not the median of n. LMS fits just over half the data and minimizes the residual for a single observation.

LTS estimator. Rousseeuw (1984, 1985) proposed the high-breakdown LTS estimator as an efficient alternative to LMS. The LTS estimator is formed by minimizing the h out of n ordered squared residuals. Rousseeuw and Leroy (1987) recommend $h = n(1-\alpha) + 1$ where α is the trimmed percentage. This estimator is attractive because α can be selected to prevent some of the poor results (efficiency) that other 50% breakdown estimators show.

M-estimator. Huber (1973) developed the *M*-estimator by minimizing a symmetric function of the residuals over the parameter estimates. These estimators are the maximum likelihood solution to $\min_{\beta} \sum_{i=1}^{n} \rho(e_i / s)$ where ρ is the residual weighting (influence) function, and s is the scale estimate to ensure that if the y values are multiplied by a constant c, then the estimated regression coefficients will also be multiplied by c. Several residual weighting functions are possible based on the downweighting philosophy (see Montgomery and Peck, 1992).

MM estimator. The *MM* estimator is a high-breakdown and high-efficiency estimator with three stages. The initial estimate is a high-breakdown estimate using an *S*-estimate. The second stage computes an *M*-estimate of the errors' scale from the initial *S*-

estimate residuals. The last step is an *M*-estimate of the regression parameters using a redescending ψ function that assigns a weight of 0.0 to large residuals.

Standard Generalized M-estimator. This estimator uses iteratively reweighted least squares to estimate the model parameters taking into account high-leverage points. The initial estimate is OLS and the estimate of scale is found by scaling the median of the absolute value of the OLS residuals. The hat diagonals are used as the measure of leverage. The *GM* objective function uses Schweppe weights that seek to improve efficiency by assigning less weight to high-leverage residuals.

Coakley and Hettmansperger estimator. This compound estimator uses LTS as the initial estimate and adjusts the estimates with empirically determined weights. The weights given to the leverage come from the robust distances using the minimum volume ellipsoid (MVE) estimator. Other components include a Schweppe-type *GM* objective function, an estimate of scale from the scaled median of the LTS residuals, the Huber ψ function and a one-step Newton-Raphson convergence approach.

Simpson and Montgomery estimator. This compound estimator uses an *S*-estimate for the initial estimate and also an *S*-estimate of scale. The scaled Krasker-Welsch weights from the *M*-estimates of covariance provide the measures of leverage. Other components include a Schweppe-type *GM* objective function, Tukey bi-weight ψ function and a one-step reweighted least squares convergence approach.

A related approach to the indirect methods from the robust regression estimators is the Rousseeuw and van Zomeren (1990) multiple outlier detection procedure. In the

original proposal, observations are classified as outliers if either the LMS residual value exceeds 2.5 or if the Mahalanobis distance measure using the MVE estimates of the mean and covariance matrix exceeds a percentile from the chi-square distribution with k degrees of freedom. The MVE estimate of the mean is the centroid of the smallest ellipse covering at least half of the observations and the estimate of the covariance matrix is determined from these cases along with a correction factor for consistency at multivariate normal distributions. Rousseeuw and van Zomeren (1991) recommend using simulated cutoff values as an update to the procedure to guard against swamping problems.

There are several published results that criticize this method for identifying too many outliers. None have used the improved genetic algorithms to compute the MVE and LMS estimates. These algorithms are computationally and statistically more efficient because more "clean" observations are used than in previous algorithms.

3.3 Monte Carlo Simulation Performance Study Planning

We use Monte Carlo simulation to test the performance of the multiple outlier detection procedures across a wide range of scenarios. The simulations generate a fixed percentage of clean observations and plant outliers at locations specified by the scenario and factor settings. The regressor variable levels for the clean observations are generated from a multivariate normal distribution with a mean of $\mu_x = 7.5$ and standard deviation of $\sigma_x = 4.0$. The choice of these parameters does not affect the results of the simulations, but is selected to be consistent with some of the results in the literature. The response for the

i^{th} clean observation is generated by $y_i = \mathbf{x}'_i\beta + \varepsilon_i$ where β is the vector of known

regression coefficients arbitrarily selected for the simulations to be 0 for the intercept and

5.0 for each of the k regressor variables and ε_i is the random error term distributed

$N(0, \sigma_e^2)$. We select σ_e^2 to be 1.00. For the planted outliers, the ith regressor variable

value for the jth observation is $x_{ij} = \bar{x}_{i,clean} + 4\delta_L + \varepsilon_{ij}^*$ where $\bar{x}_{i,clean}$ is the average of the

clean values for the ith regressor, δ_L is the magnitude of the outlying shift distance in X-

space in standard deviation units, σ_x, and ε_{ij}^* is a random variate from a Uniform (0, 0.25).

We use the ε_{ij}^* term to separate multiple observations in a cloud to protect against singular

matrices. If the ith observation is a regression outlier, the response value is calculated by

$y_i = \mathbf{x}'_i\beta + \delta_R$ where δ_R is the magnitude of the outlying distance off the regression plane in

standard deviation units, σ_e.

Where practical, simulation studies use factorial designs to characterize the effects

of specific factors on the two primary measures of performance: detection capability and

false alarm rate. The false alarm rate is the probability that a clean observation is

swamped and the complement of detection probability is the masking probability. The

factors considered are the dimension of the data, the percentage of outliers, the magnitude

of unusualness in X-space, δ_L, the magnitude of unusualness in residual, δ_R, the number of

multiple point clouds, and the proportion of regressor variables with extreme values.

The factor levels are selected to develop challenging scenarios used to discriminate

the performance of the procedures. Extensive pilot studies were run to discover the best

levels to not only challenge the procedures, but also to ensure that at least one of the candidates has detection capability for most combinations of the selected factor levels. The levels for dimension are either $k = 2$ variables with $n = 40$ observations or $k = 6$ variables with $n = 60$ observations. The levels for the percentage of outliers are typically 10% and 20%, although some studies vary these factor settings. The levels for the outlying distances δ_L and δ_R are typically between 3 and 5 standard deviation units. The number of clouds is selected as a factor with settings of usually 1 or 2, because the most difficult outlier configuration is a mean shift with a single cloud of observations that are clustered close to one another, yet not replicated (Rocke and Woodruff, 1996). The levels for the number of outlying variables are either all k variables, as commonly seen in the literature, one of the k variables, or 3 of 6 variables.

To properly and fairly compare the methods, we set their parameters such that the expected false alarm probability is 5% under the null hypothesis of no outliers. For example, the simulated cutoff value for an indirect robust regression procedure is calculated as the 95th percentile of the absolute value of the residuals from clean data (no planted outliers).

The Monte Carlo simulations are all performed in *S-Plus* (the simulation and procedure code is shown in Appendix A) and are classified into two main categories of regression outliers: 1) interior X-space outliers and 2) exterior X-space outliers. Studies are made within each of these main categories to best evaluate the procedures across a wide range of possible regression scenarios. Figure 3.1 displays the appropriate section numbers within the chapter for the study results.

```
                        ┌──────────┐
                        │ Multiple │
                        │ Outliers │
                        └──────────┘
              ┌───────────────┴───────────────┐
       ┌──────────────┐               ┌──────────────┐
       │ Interior X-space │           │ Exterior X-space │
       │ regression outliers │        │ regression outliers │
       │     3.4.1     │               │     3.4.2     │
       └──────────────┘               └──────────────┘
        ┌──────┴──────┐               ┌──────┴──────┐
  ┌──────────┐ ┌──────────┐    ┌──────────┐ ┌──────────┐
  │ Multiple │ │ Randomly │    │ Not unusual │ │ Unusual in │
  │ point clouds │ │ scattered outliers │ │ in Y-space │ │ Y-space │
  │          │ │  3.4.1.1 │    │          │ │          │
  └──────────┘ └──────────┘    └──────────┘ └──────────┘
```
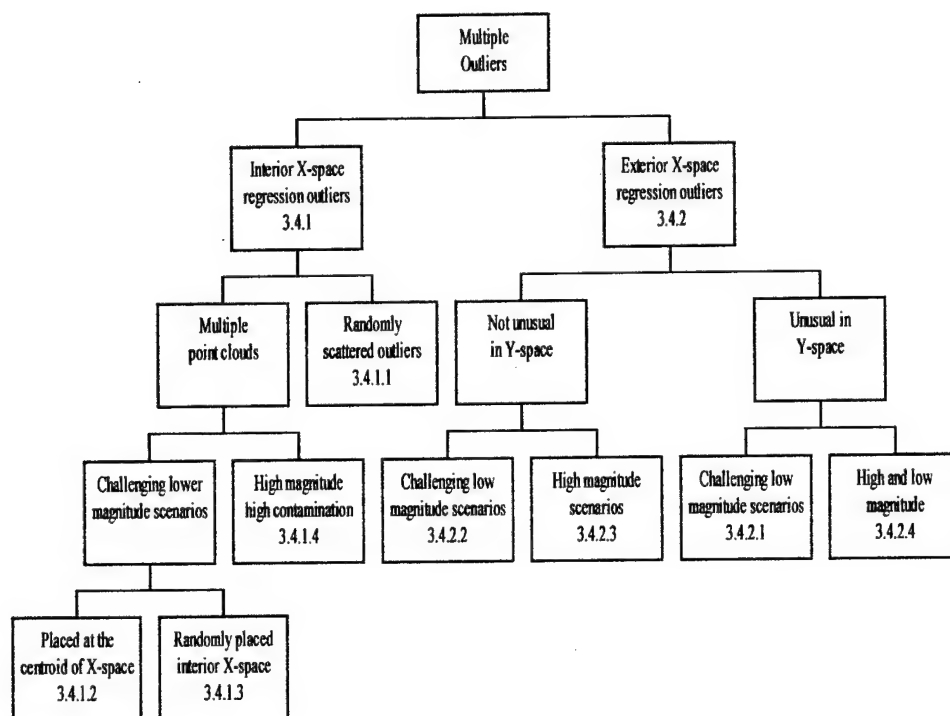
Figure 3.1. Organization chart for the Monte Carlo simulation studies.

3.4 Performance Study Results

Each procedure's performance is evaluated on its ability to detect the planted

outliers and avoid false alarms. Both the detection capability and false alarm rate (shown

in parentheses in the tables) are reported for 500 replications. Common random numbers

ensure that each procedure evaluates the same 500 sets of data. Section 3.4.1 describes

the experiment designs, results and performance summaries for interior X-space regression

outliers. The high-leverage (exterior X-space) regression outlier studies are in Section

3.4.2.

3.4.1 Interior X-space Regression Outliers

This set of experiments evaluates the ability of the methods to identify regression outliers when all regressor variable values are not unusual in X-space. That is, there are no high-leverage points intentionally planted in the samples. The response values for the interior X-space outlying observations are offset a distance δ_R from the regression plane obtained from the clean cases. There are three studies in this section based on the configuration of the outliers. In the first study, the multiple outliers are randomly scattered in the interior of X-space. The second study considers multiple point clouds or clusters of outliers that are located near the centroid of X-space. The third study considers multiple point clouds randomly placed (different for each replication) in the interior of X-space. The measures of performance are the probability of detection and the probability that a clean observation is incorrectly classified as an outlier. The average value of these probabilities and the active effects from the analysis of variance are displayed in the last rows of the tables to provide summary information on the techniques.

The direct procedures evaluated in these studies and the accompanying abbreviations for the tables of results are: 1) the Sebert et al. clustering algorithm (SM&R), 2) the Swallow and Kianifard (S&K) recursive residual algorithm, 3) the Pena and Yohai (P&Y) influence matrix algorithm, and 4) the Hadi and Simonoff sequential point addition algorithm. Both the original Hadi and Simonoff algorithm (HS93) and the updated version (HS97) are considered. The selected indirect procedures that incorporate the residuals from regression estimators are OLS, LMS, LTS, M and MM. To limit the

scope, residuals from compound estimators and *GM*-estimators are not considered for these studies because we are not considering high-leverage points.

3.4.1.1 Randomly Scattered Regression Outliers in the Interior of X-Space

This study evaluates performance when the outliers have random levels of the regressor variables with the same distribution as the clean observations but the response values are placed at a specified distance δ_R off the regression plane. The response to the ith clean case is generated by $y_i = \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i$ where $\boldsymbol{\beta}$ is the vector of known regression coefficients selected for the simulations to be 0 for the intercept and 5.0 for each of the k regressor variables, $\mathbf{x}_i$ is the vector of levels for the k regressor variables distributed multivariate normal with mean 7.5 and standard deviation 4.0 and ε_i is the random error term distributed $N(0, \sigma_e^2)$ with σ_e^2 set to 1.00. The response to the ith outlying observation is generated by $y_i = \mathbf{x}_i'\boldsymbol{\beta} + \delta_R$ where δ_R is the outlying distance off the regression plane in standard deviation units of σ_e. The design in Table 3.1 considers dimension, density of outliers (as a percentage of the sample size), and δ_R as the effects. The probability of correctly identifying the known outliers and also the false alarm probability in parentheses are the results from the Monte Carlo simulations.

The OLS, *M* and *MM* regression estimators' detection capability stands out in the resulting probabilities reported in Table 3.1. These indirect methods are the only ones with any power at a magnitude of $\delta_R = 3\sigma_e$ and they have nearly perfect detection capability at $4\sigma_e$ and beyond. Although it has excellent detection capability, the OLS

Table 3.1. Design matrix with detection capability and false alarm rates (in parentheses) for regression outliers generated from random levels of the regressor variables from the interior of X-space.

A n,k	B dens	C δ_R	HS93 (0.05)	HS93 (0.20)	HS97 (0.05)	HS97 (0.20)	S&K	P&Y	SM&R	OLS	M	MM	LTS	LMS
40, 2	10%	3σ	0.108 (.001)	0.453 (.011)	0.105 (.001)	0.415 (.013)	0.450 (.010)	0.220 (.020)	0.820 (.113)	0.964 (.056)	0.978 (.061)	0.989 (.060)	0.581 (.012)	0.503 (.012)
60, 6	10%	3σ	0.007 (.001)	0.165 (.005)	0.001 (.007)	0.107 (.005)	0.292 (.008)	0.140 (.012)	0.691 (.118)	0.867 (.053)	0.910 (.058)	0.918 (.057)	0.210 (.008)	0.471 (.025)
40, 2	20%	3σ	0.070 (.001)	0.436 (.014)	0.071 (.001)	0.260 (.007)	0.244 (.011)	0.123 (.018)	0.739 (.136)	0.851 (.084)	0.890 (.090)	0.938 (.085)	0.538 (.025)	0.464 (.019)
60, 6	20%	3σ	0.000 (.000)	0.107 (.003)	0.000 (.000)	0.037 (.002)	0.108 (.008)	0.042 (.009)	0.520 (.136)	0.683 (.085)	0.740 (.092)	0.770 (.088)	0.225 (.017)	0.445 (.036)
40, 2	10%	4σ	0.656 (.003)	0.945 (.012)	0.653 (.003)	0.973 (.016)	0.891 (.011)	0.435 (.021)	0.979 (.076)	0.996 (.064)	1.000 (.058)	1.000 (.058)	0.974 (.017)	0.953 (.011)
60, 6	10%	4σ	0.312 (.001)	0.927 (.007)	0.312 (.001)	0.805 (.007)	0.842 (.008)	0.345 (.017)	0.928 (.094)	0.988 (.067)	0.997 (.054)	0.998 (.054)	0.726 (.007)	0.890 (.020)
40, 2	20%	4σ	0.529 (.003)	0.909 (.016)	0.478 (.003)	0.704 (.016)	0.727 (.019)	0.209 (.013)	0.933 (.104)	0.982 (.118)	0.987 (.096)	0.993 (.086)	0.970 (.016)	0.942 (.013)
60, 6	20%	4σ	0.208 (.001)	0.803 (.007)	0.188 (.001)	0.518 (.004)	0.535 (.013)	0.083 (.006)	0.754 (.112)	0.941 (.123)	0.952 (.109)	0.961 (.099)	0.730 (.007)	0.889 (.021)
40, 2	10%	5σ	0.985 (.003)	1.000 (.012)	0.985 (.003)	1.000 (.015)	0.989 (.013)	0.618 (.019)	0.996 (.051)	1.000 (.076)	1.000 (.055)	1.000 (.053)	0.995 (.016)	0.991 (.012)
60, 6	10%	5σ	0.960 (.002)	1.000 (.010)	0.967 (.002)	0.992 (.007)	0.996 (.009)	0.521 (.015)	0.980 (.076)	1.000 (.085)	1.000 (.050)	1.000 (.049)	0.953 (.005)	0.987 (.019)
40, 2	20%	5σ	0.936 (.003)	1.000 (.016)	0.916 (.003)	0.974 (.014)	0.966 (.032)	0.334 (.012)	0.980 (.069)	0.996 (.168)	0.998 (.088)	1.000 (.074)	0.997 (.016)	0.996 (.012)
60, 6	20%	5σ	0.778 (.002)	0.968 (.009)	0.745 (.002)	0.908 (.005)	0.850 (.013)	0.137 (.008)	0.867 (.090)	0.983 (.173)	0.985 (.113)	0.993 (.092)	0.966 (.006)	0.990 (.018)
Average probabilities			0.462 (.002)	0.726 (.010)	0.452 (.002)	0.641 (.009)	0.658 (.013)	0.267 (.014)	0.849 (.098)	0.938 (.096)	0.953 (.077)	0.963 (.071)	0.739 (.013)	0.793 (.018)
Significant effects detection capability			A, C	A, C	C	C	B, C	A, B, C, BC	A, B, C	C	C	C	A, C	C
Significant effects false alarms			A, C	A, C	A, B	A, B	A, B, C	A, B, C, BC	A, B, C	B, C, BC	B	B	A	A, AC

method is unsatisfactory because it swamps clean observations as indicated by the high false alarm rate. This is attributed to a degradation in parameter estimates (that worsen as a function of δ_R) such that the clean data are no longer fit well. The M-estimator has some difficulty with false alarms in high-density scenarios as expected and the MM procedure has a slightly lower, although still high, false alarm rate. The high-breakdown methods of LMS and LTS are preferred for outlying magnitudes at $4\sigma_e$ and beyond because of the competitive detection probabilities and low false alarm rates. The LMS estimator is slightly preferred over the LTS in low-density scenarios and the opposite is true for the high-density scenarios.

For the direct methods, the Pena and Yohai method is significantly outperformed by all other techniques at these outlying distances. Further simulation demonstrates the algorithm does have much better detection capability if δ_R is greater than approximately $7\sigma_e$. The Sebert et al. clustering procedure has decent detection capability, but suffers from a large false alarm probability in many scenarios. The false alarm rate for both Hadi and Simonoff versions is abnormally low and the detection capability is also low at $4\sigma_e$ and below. This presented an opportunity to increase detection capability by decreasing the cutoff value from the t distribution based on a Bonferroni approach. The value of α is increased from 0.05 to 0.20. The results in Table 3.1 do indicate a greater detection capability is possible without severe impact to false alarm probabilities. We note that the original Hadi and Simonoff version from 1993 has nearly identical performance to the

improved version from 1997 for $\alpha = 0.05$. The 1993 version moderately outperforms the improved version for $\alpha = 0.20$.

3.4.1.2 Regression Outliers in Multiple Point Clouds at the Centroid of X-Space

This study evaluates the performance of the procedures when there are multiple observations forming one or two clusters at the centroid of X-space that are off the regression plane. As usual, the response value for the ith clean observation is generated by $y_i = \mathbf{x}'_{it}\boldsymbol{\beta} + \varepsilon_i$ where $\boldsymbol{\beta}$ is the vector of known regression coefficients selected for the simulations to be 0 for the intercept and 5.0 for each of the k regressor variables, $\mathbf{x}_i$ is the vector of levels for the k regressor variables distributed multivariate normal with mean 7.5 and standard deviation 4.0 and ε_i is the random error term distributed $N(0, \sigma_e^2)$ with σ_e^2 set to 1.00. The response value for the ith outlying case is generated by $y_i = \mathbf{x}'_{it}\boldsymbol{\beta} + \delta_R$ where $\mathbf{x}_i$ is the vector of k regressor variables distributed Uniform (7.375, 7.625) and δ_R is the outlying distance off the regression plane in standard deviation units. If there are two clouds, the response values for the outliers in the first cloud are generated as above and the second cloud's response values are generated by $y_i = \mathbf{x}'_{it}\boldsymbol{\beta} - \delta_R$. The four factors for this experiment are dimension, outlier density, outlying distance (δ_R), and the number of clouds. The design and results are displayed in Table 3.2. For this particular study, the levels of δ_R are close to one another because initial experimentation indicated that none of the procedures had detection capability below $3\sigma_e$ and nearly all had virtually perfect detection capability at $5\sigma_e$ and beyond.

Table 3.2. Design matrix with detection capability and false alarm rates (in parentheses) for regression outliers in multiple point clouds at the centroid of X-space. A single cloud is placed δ_R off the regression surface.

A n, k	B dens	C δ_R	D cld	HS93	HS97	S & K	P & Y	SM&R	OLS	M	MM	LTS	LMS
40, 2	10%	3σ	1	0.187 (.001)	0.200 (.001)	0.503 (.015)	0.100 (.028)	0.930 (.138)	1.000 (.054)	1.000 (.060)	1.000 (.059)	0.605 (.014)	0.469 (.011)
60, 6	10%	3σ	1	0.064 (.001)	0.080 (.001)	0.484 (.010)	0.010 (.027)	0.970 (.143)	1.000 (.049)	1.000 (.055)	1.000 (.056)	0.020 (.007)	0.494 (.022)
40, 2	20%	3σ	1	0.143 (.004)	0.060 (.001)	0.291 (.016)	0.265 (.041)	0.910 (.176)	0.995 (.081)	1.000 (.087)	1.000 (.083)	0.622 (.025)	0.456 (.026)
60, 6	20%	3σ	1	0.117 (.001)	0.000 (.001)	0.219 (.013)	0.090 (.033)	0.940 (.183)	0.995 (.078)	1.000 (.082)	1.000 (.079)	0.040 (.027)	0.423 (.075)
40, 2	10%	4σ	1	0.832 (.002)	0.850 (.006)	0.934 (.016)	0.345 (.027)	1.000 (.095)	1.000 (.063)	1.000 (.053)	1.000 (.055)	1.000 (.014)	0.995 (.011)
60, 6	10%	4σ	1	0.756 (.002)	0.840 (.003)	0.967 (.012)	0.125 (.027)	0.995 (.105)	1.000 (.059)	1.000 (.052)	1.000 (.052)	0.951 (.006)	1.000 (.022)
40, 2	20%	4σ	1	0.782 (.006)	0.370 (.001)	0.814 (.022)	0.635 (.033)	0.990 (.144)	1.000 (.112)	1.000 (.086)	1.000 (.083)	0.999 (.089)	0.995 (.011)
60, 6	20%	4σ	1	0.805 (.003)	0.020 (.005)	0.829 (.018)	0.410 (.037)	1.000 (.152)	1.000 (.112)	1.000 (.049)	1.000 (.081)	1.000 (.014)	0.949 (.006)
40, 2	10%	3σ	2	0.116 (.000)	0.150 (.001)	0.529 (.008)	0.043 (.022)	0.900 (.107)	1.000 (.040)	1.000 (.052)	1.000 (.052)	0.648 (.015)	0.484 (.011)
60, 6	10%	3σ	2	0.031 (.027)	0.041 (.000)	0.531 (.006)	0.008 (.022)	0.950 (.117)	1.000 (.037)	1.000 (.051)	1.000 (.050)	0.017 (.008)	0.493 (.022)
40, 2	20%	3σ	2	0.122 (.004)	0.100 (.004)	0.293 (.002)	0.050 (.039)	0.835 (.145)	1.000 (.038)	1.000 (.052)	1.000 (.049)	0.656 (.015)	0.478 (.009)
60, 6	20%	3σ	2	0.073 (.001)	0.070 (.001)	0.200 (.001)	0.011 (.041)	0.940 (.154)	1.000 (.035)	1.000 (.045)	1.000 (.045)	0.030 (.050)	0.480 (.020)
40, 2	10%	4σ	2	0.755 (.002)	0.810 (.003)	0.906 (.016)	0.140 (.032)	0.990 (.057)	1.000 (.040)	1.000 (.052)	1.000 (.052)	1.000 (.014)	0.998 (.011)
60, 6	10%	4σ	2	0.705 (.002)	0.781 (.002)	0.973 (.012)	0.025 (.024)	0.998 (.065)	1.000 (.037)	1.000 (.050)	1.000 (.049)	0.952 (.006)	1.000 (.022)
40, 2	20%	4σ	2	0.734 (.006)	0.803 (.006)	0.831 (.018)	0.188 (.045)	0.980 (.090)	1.000 (.038)	1.000 (.051)	1.000 (.050)	0.999 (.013)	0.999 (.009)
60, 6	20%	4σ	2	0.803 (.003)	0.840 (.003)	0.807 (.018)	0.065 (.047)	0.995 (.106)	1.000 (.035)	1.000 (.045)	1.000 (.044)	0.938 (.005)	0.998 (.021)
Average probabilities				0.439 (.004)	0.376 (.003)	0.632 (.013)	0.157 (.033)	0.958 (.124)	0.999 (.057)	1.000 (.058)	1.000 (.059)	0.655 (.020)	0.732 (.019)
Significant effects Detection capability				C	C	B, C, BC	A, C, D, AC, BD	C	none	none	none	A, C, AC	C
Significant effects False alarms				none	B, AD	C, D, CD	B	A, B, C, D, CD	A, B, C, D, BD	D, BD	A, B, D, BD	None	none

The methods are more successful at detecting these outlying observations in clouds at the centroid of X-space compared to similar scenarios with randomly scattered outliers in Section 3.4.1.1. Again, the OLS, *M* and *MM* indirect methods are superior in detection capability. OLS has problems with swamping if there is a single cloud for the reasons described in Section 3.4.1.1. However, when there are two clouds, there is no swamping because there is an equal and opposite "pull" on the regression surface from each cloud that leaves the parameter estimates essentially unchanged from those obtained with clean observations only. *M* and *MM* have nearly identical detection and false alarm probabilities. Except for the two highlighted scenarios, the Hadi and Simonoff 1997 updated procedure performs as well as or slightly better than the original 1993 version. All other methods have consistent results with Section 3.4.1.1.

3.4.1.3 Regression Outliers in Multiple Point Clouds: Regressor Variables Randomly Scattered on the Interior of X-Space

The multiple outlier clouds for this section are placed at different locations in X-space rather than the centroid for each replication. The location of the regressors for outlying observations in a single point cloud is determined by using the median of the first three clean observations for each variable. The regressor variables for the outlying observations then vary as Uniform (0, 0.25) around this median value. Outliers in a second cloud, if applicable, vary around the median value of the last three clean observations in each variable. Recall that each regressor variable for the clean observations is distributed $N(7.5, 4^2)$. We found that using the median of three

observations provides adequate coverage of interior X-space and that more than three observations tends to place the outlying observations too close to the centroid of X-space. The response values are found exactly as in the previous two sections with three levels of δ_R specified as the outlying magnitude. The factors for this experiment design in Table 3.3 are dimension, contamination, number of clouds and the outlying distance δ_R.

The results in Table 3.3 indicate the findings are consistent with the first two studies except this scenario is more challenging. Most main effects and many two factor interactions are significant for detection capability except for the high breakdown regression estimators. The least squares estimates do not fit the outlying cloud(s) well as evidenced by the high probability of detection; however, they do chase these observations enough to swamp some clean observations. The M and MM estimators have moderately better detection probabilities than OLS and significantly better false alarm rates, although well above nominal levels in the high-dimension, high-density scenarios. The high-breakdown methods are not impacted with high false alarms and reliably detect the outliers at $4\sigma_e$ and beyond. Sebert et al. is no longer competitive with the other procedures because of a consistent high false alarm problem and decreased power. Pena and Yohai has slightly better performance with the increased leverage for these outlying clouds, although still not competitive with any other procedure. Both Hadi and Simonoff procedures have very low false alarm rates and further testing demonstrates substantial improvement in detection capability is possible if α is increased to as much as 0.30.

Table 3.3. Design matrix with detection capability and false alarm rates (in parentheses) for regression outliers with the regressor variables for the outliers determined from the median of the first three clean observations for each variable. In the case of two clouds, the regressor variables for the second cloud are determined from the median of the last three clean observations.

A n, k	B dens	C δ_R	D cld	HS93	HS97	S & K	P & Y	SM&R	OLS	M	MM	LTS	LMS
40, 2	10%	3σ	1	0.140 (.002)	0.140 (.002)	0.439 (.012)	0.165 (.028)	0.598 (.149)	0.693 (.059)	0.770 (.063)	0.780 (.061)	0.523 (.015)	0.476 (.009)
60, 6	10%	3σ	1	0.029 (.000)	0.023 (.000)	0.337 (.008)	0.101 (.025)	0.395 (.155)	0.590 (.058)	0.705 (.060)	0.724 (.058)	0.269 (.008)	0.460 (.021)
40, 2	20%	3σ	1	0.000 (.013)	0.013 (.000)	0.174 (.008)	0.151 (.038)	0.260 (.203)	0.519 (.094)	0.588 (.095)	0.646 (.085)	0.455 (.029)	0.405 (.024)
60, 6	20%	3σ	1	0.000 (.000)	0.002 (.000)	0.043 (.008)	0.019 (.039)	0.097 (.179)	0.366 (.101)	0.424 (.108)	0.456 (.107)	0.216 (.027)	0.300 (.063)
40, 2	10%	4σ	1	0.445 (.002)	0.445 (.002)	0.754 (.012)	0.465 (.037)	0.825 (.130)	0.903 (.071)	0.951 (.060)	0.951 (.062)	0.825 (.015)	0.798 (.008)
60, 6	10%	4σ	1	0.202 (.000)	0.202 (.000)	0.658 (.009)	0.348 (.026)	0.643 (.145)	0.826 (.074)	0.943 (.059)	0.958 (.055)	0.647 (.006)	0.813 (.019)
40, 2	20%	4σ	1	0.198 (.013)	0.129 (.002)	0.391 (.012)	0.357 (.033)	0.494 (.199)	0.762 (.136)	0.843 (.116)	0.898 (.093)	0.807 (.015)	0.774 (.015)
60, 6	20%	4σ	1	0.048 (.000)	0.002 (.000)	0.078 (.008)	0.079 (.037)	0.172 (.190)	0.572 (.150)	0.628 (.150)	0.695 (.132)	0.552 (.022)	0.700 (.045)
40, 2	10%	5σ	1	0.858 (.019)	0.858 (.002)	0.929 (.014)	0.728 (.034)	0.939 (.099)	0.984 (.084)	0.946 (.058)	0.996 (.057)	0.965 (.013)	0.951 (.009)
60, 6	10%	5σ	1	0.755 (.001)	0.745 (.001)	0.904 (.010)	0.741 (.029)	0.780 (.141)	0.953 (.097)	0.994 (.057)	0.994 (.050)	0.903 (.006)	0.960 (.019)
40, 2	20%	5σ	1	0.605 (.002)	0.464 (.002)	0.625 (.018)	0.672 (.035)	0.746 (.179)	0.896 (.195)	0.953 (.133)	0.982 (.083)	0.968 (.013)	0.953 (.012)
60, 6	20%	5σ	1	0.408 (.003)	0.113 (.000)	0.199 (.013)	0.274 (.033)	0.297 (.195)	0.755 (.205)	0.803 (.189)	0.875 (.138)	0.874 (.010)	0.923 (.026)

Table 3.3 (cont).

40, 2	10%	3σ	2	0.165 (.001)	0.165 (.001)	0.499 (.014)	0.136 (.023)	0.616 (.132)	0.736 (.057)	0.790 (.061)	0.797 (.060)	0.531 (.014)	0.478 (.008)
60, 6	10%	3σ	2	0.052 (.000)	0.052 (.000)	0.417 (.009)	0.093 (.021)	0.565 (.135)	0.688 (.053)	0.773 (.056)	0.776 (.055)	0.280 (.009)	0.478 (.021)
40, 2	20%	3σ	2	0.018 (.001)	0.018 (.001)	0.271 (.008)	0.060 (.023)	0.404 (.174)	0.577 (.087)	0.661 (.085)	0.709 (.080)	0.493 (.016)	0.438 (.015)
60, 6	20%	3σ	2	0.000 (.000)	0.000 (.000)	0.106 (.006)	0.005 (.025)	0.258 (.189)	0.494 (.093)	0.561 (.095)	0.598 (.089)	0.260 (.015)	0.400 (.041)
40, 2	10%	4σ	2	0.478 (.002)	0.470 (.002)	0.779 (.016)	0.343 (.036)	0.871 (.107)	0.935 (.067)	0.960 (.060)	0.960 (.060)	0.821 (.014)	0.784 (.008)
60, 6	10%	4σ	2	0.260 (.000)	0.260 (.000)	0.757 (.009)	0.243 (.019)	0.803 (.118)	0.909 (.067)	0.961 (.053)	0.960 (.053)	0.654 (.006)	0.818 (.019)
40, 2	20%	4σ	2	0.191 (.001)	0.191 (.001)	0.581 (.013)	0.218 (.024)	0.636 (.148)	0.828 (.122)	0.899 (.097)	0.933 (.083)	0.805 (.013)	0.782 (.011)
60, 6	20%	4σ	2	0.049 (.000)	0.039 (.000)	0.298 (.009)	0.110 (.025)	0.428 (.183)	0.727 (.142)	0.811 (.121)	0.856 (.100)	0.613 (.010)	0.797 (.023)
40, 2	10%	5σ	2	0.846 (.002)	0.843 (.002)	0.953 (.019)	0.643 (.021)	0.954 (.073)	0.991 (.081)	0.998 (.057)	0.998 (.056)	0.966 (.014)	0.955 (.008)
60, 6	10%	5σ	2	0.795 (.001)	0.795 (.001)	0.945 (.011)	0.569 (.021)	0.947 (.105)	0.986 (.087)	0.997 (.049)	0.999 (.049)	0.904 (.005)	0.961 (.019)
40, 2	20%	5σ	2	0.583 (.002)	0.580 (.002)	0.850 (.021)	0.513 (.020)	0.837 (.124)	0.949 (.177)	0.984 (.094)	0.994 (.073)	0.971 (.012)	0.953 (.011)
60, 6	20%	5σ	2	0.483 (.001)	0.308 (.000)	0.525 (.014)	0.272 (.025)	0.584 (.185)	0.888 (.197)	0.941 (.137)	0.965 (.092)	0.900 (.007)	0.958 (.018)
Average probabilities				0.317 (.003)	0.286 (.001)	0.521 (.012)	0.304 (.028)	0.590 (.152)	0.772 (.106)	0.829 (.088)	0.854 (.076)	0.675 (.013)	0.721 (.020)
Significant effects Detection capability				A, B, C	A, B, C, BC	A, B, C, D, AB	A, B, C, D	A,B,C,D, AB, AD	A, B, C, D	A, B, C, D, AB	A, B, C	A, C	C
Significant effects false alarms				A, D	A, B, C	A, B, D, AB	D, AB	A, B, C, D	A, B, C, D, BC	B, AB, BC	A, B, D, AB	B, C, AC	A, B

82

3.4.1.4 Supplemental Runs of Higher Outlying Distances and Outlier Density

This study examines the effect of increasing the factor settings for δ_R, the outlying

distance off the regression plane, and the contamination or percentage of outliers. The

factor level for δ_R is changed from the usual challenging $3 - 5\sigma_e$ range to a low level of $5\sigma_e$

and a high level of $10\sigma_e$. The percentage contamination is changed to 15% for the low

level and 30% for the high level. The first two scenarios in Table 3.4 are randomly

scattered outliers as investigated in Section 3.4.1.1. The next four scenarios have multiple

point clouds placed at or near the centroid. The last 8 scenarios form a 2^{4-1} fractional

factorial with the outliers placed in clouds randomly throughout X-space similar to those

of Section 3.4.1.3.

These runs produce some rather different results from the preceding studies. With

the contamination set higher, we now see more clearly that the OLS and M estimators

break down by the astronomical false alarm rates. The MM estimator reliably detects the

planted outliers (the one exception is the second shaded scenario). The original Hadi and

Simonoff procedure is superior to the modified version for these scenarios. The first

shaded scenario is different from most because of the failure of the high breakdown

regression estimators and the discrepancy in performance between the two Hadi and

Simonoff versions. All procedures fail in the second shaded scenario. In most instances,

the LMS, LTS and MM estimators perform the best when accounting for the false alarm

rates. We also note the success of the Pena and Yohai procedure for the low-density, high

outlying distance scenarios. Surprisingly, the only active effect in this operating region for

Table 3.4. Design matrix with detection capability and false alarm rates (in parentheses) for high-magnitude, high-density runs for regression outliers in the interior of X-space.

A n, k	B clds	C density	D δ_R	HS93	HS97	S&K	P&Y	SM&R	OLS	M	MM	LTS	LMS
60, 6	Rand	15%	10σ	1.000 (.002)	1.000 (.002)	1.000 (.021)	0.728 (.007)	0.990 (.056)	1.000 (.349)	1.000 (.048)	1.000 (.047)	1.000 (.005)	1.000 (.018)
60, 6	Rand	30%	10σ	0.843 (.002)	0.539 (.000)	0.704 (.019)	0.036 (.003)	0.941 (.085)	0.997 (.688)	0.996 (.642)	1.000 (.055)	1.000 (.004)	1.000 (.012)
60, 6	1	15%	10σ	0.992 (.001)	0.969 (.001)	1.000 (.062)	1.000 (.019)	1.000 (.046)	1.000 (.320)	1.000 (.043)	1.000 (.043)	1.000 (.006)	1.000 (.018)
60, 6	1	30%	10σ	0.956 (.007)	0.000 (.173)	0.983 (.020)	0.000 (.032)	1.000 (.161)	1.000 (.734)	1.000 (.702)	1.000 (.041)	0.950 (.035)	0.980 (.021)
60, 6*	1	15%	10σ	0.923 (.003)	0.240 (.001)	1.000 (.010)	1.000 (.035)	1.000 (.043)	1.000 (.375)	1.000 (.044)	1.000 (.043)	1.000 (.007)	1.000 (.018)
60, 6*	1	30%	10σ	0.817 (.046)	0.010 (.045)	0.090 (.108)	0.000 (.054)	1.000 (.165)	1.000 (.578)	1.000 (.598)	0.980 (.058)	0.120 (.447)	0.540 (.243)
40, 2	1	15%	5σ	0.733 (.002)	0.593 (.000)	0.917 (.019)	0.728 (.026)	0.882 (.161)	0.978 (.143)	0.995 (.063)	0.998 (.059)	0.973 (.011)	0.963 (.009)
60, 6	1	15%	10σ	0.950 (.001)	0.720 (.001)	0.991 (.020)	0.995 (.029)	0.894 (.154)	0.999 (.389)	1.000 (.065)	1.000 (.047)	1.000 (.005)	1.000 (.020)
40, 2	2	15%	10σ	1.000 (.002)	1.000 (.001)	1.000 (.065)	0.998 (.014)	0.998 (.047)	1.000 (.339)	1.000 (.054)	1.000 (.053)	1.000 (.012)	1.000 (.009)
60, 6	2	15%	5σ	0.635 (.001)	0.534 (.000)	0.853 (.011)	0.412 (.021)	0.804 (.134)	0.948 (.134)	0.993 (.064)	0.996 (.058)	0.914 (.004)	0.963 (.018)
40, 2	1	30%	10σ	0.837 (.009)	0.344 (.000)	0.606 (.080)	0.000 (.043)	0.950 (.199)	0.990 (.631)	0.987 (.627)	0.993 (.059)	0.980 (.022)	0.980 (.018)
60, 6	1	30%	5σ	0.248 (.001)	0.054 (.000)	0.013 (.012)	0.000 (.054)	0.133 (.267)	0.562 (.293)	0.554 (.315)	0.599 (.310)	0.405 (.155)	0.434 (.188)
40, 2	2	30%	5σ	0.473 (.006)	0.238 (.000)	0.309 (.015)	0.033 (.023)	0.739 (.152)	0.855 (.295)	0.875 (.291)	0.948 (.191)	0.945 (.022)	0.925 (.024)
60, 6	2	30%	10σ	0.805 (.002)	0.124 (.000)	0.158 (.009)	0.040 (.036)	0.688 (.282)	0.996 (.609)	0.993 (.613)	0.990 (.049)	0.980 (.013)	0.980 (.028)
Average probabilities				0.801 (.006)	0.455 (.016)	0.687 (.034)	0.426 (.028)	0.859 (.139)	0.952 (.420)	0.957 (.298)	0.965 (.080)	0.876 (.053)	0.912 (.046)
Significant effects Detection capability				C, D	C	C	C	none	none	none	none	none	none
Significant effects False alarms				none	C, D	none	none	C	C, D	C, D, AB	none	none	none

*The outliers for these scenarios are placed at 0.5σ from the centroid of X-space and the two above are placed approximately at the centroid.

detection capability from the ½ fraction design is the percentage of outliers; this is true only for the direct procedures.

3.4.2 Exterior X-space Regression Outliers

This section evaluates a method's ability to detect observations outlying in X-space (high-leverage) and also off the regression plane (residual outliers). The same direct procedures are evaluated. We change the indirect methods because of known vulnerabilities of the M and MM estimators in high-leverage situations. The indirect procedures are the bounded influence generalized M-estimator (GM) and the compound robust regression estimators of Coakley and Hettmansperger (CE C&H) and Simpson and Montgomery (CE S&M). We also test the procedures from Rousseeuw and van Zomeren (1990, 1991) that suggest the MVE robust distances to identify observations remote in X-space and LMS standardized residuals to find regression outliers. The original proposal (R&vZ chi) uses robust distance cutoff values from percentiles of the chi-square distribution ($\chi^2_{k,0.975}$) and a rule of thumb cutoff value for the LMS standardized residuals (2.5). Their subsequent recommendation (R&vZ sim) uses simulated cutoff values.

The first study, Section 3.4.2.1, has multiple point clouds at various leverage locations. The regressor variable values are remote in all k regressors for the planted outliers as is often reported in the literature. Section 3.4.2.2 is similar to the first study but ensures the response values for the outliers, although off the regression surface, are not unusual with respect to the clean responses. That is, the regression outliers are Y-

space inliers. We next investigate in Section 3.4.2.3 the effect if the outliers are not unusual in all k regressor variables and the outlying magnitude is increased. The last Section 3.4.2.4 evaluates the performance when there is a remote cloud in X-space of regression outliers and also other regression outliers in the interior of X-space. This last experiment looks at the possibility of the high-leverage, large-magnitude regression outliers masking the low-leverage smaller magnitude regression inliers. Throughout all of the studies, cutoff values and other internal parameters are selected for each procedure to ensure the expected false alarm rate is approximately 0.05 under the null hypothesis of no outliers.

3.4.2.1 Regression Outliers in Clouds that are Unusual in X-space for All Regressors

This study evaluates performance for scenarios with high-leverage multiple point clouds that are off the regression plane. The scenarios are similar to those used by Sebert et al., Hadi and Simonoff, and Kianifard and Swallow. The regressor and response values for the clean observations are computed as described in Section 3.4.1. The value of the ith regressor variable for the jth planted outlier is $x_{ij} = \overline{x}_{i,clean} + \delta_L + \varepsilon_{ij}^*$ where $\overline{x}_{i,clean}$ is the average of the clean observations for the ith regressor variable, δ_L is the magnitude of the outlying distance in X-space in standard deviation units, σ_x, and ε_{ij}^* is a random variate from a Uniform $(0, 0.25)$ distribution. In this section, all k regressor variable values for the outlying observations are generated as above. A multiple point cloud is placed at the edge of interior X-space when the leverage magnitude is at the low factor setting ($\delta_L =$

$2\sigma_x$). The cloud is significantly remote in X-space for the high factor setting of leverage magnitude ($\delta_L = 5\sigma_x$). If there are 2 clouds, the second cloud is placed at approximately the same location in X-space but the response value is $2\sigma_e$ above that of the first cloud. As an example, for $k = 2$, leverage magnitude $\delta_L = 2\sigma_x$, and residual magnitude $\delta_R = 5\sigma_e$, the regressor variable values for the ith outlying observation in either cloud are $x_1 = 7.5 + 2(4) + \varepsilon_{ij}^*$ and $x_2 = 7.5 + 2(4) + \varepsilon_{ij}^*$. The response value for the ith outlying observation in the first cloud is calculated as $y_i = 5x_{1i} + 5x_{2i} + 5$ and the ith response value in the second cloud is calculated as $y_i = 5x_{1i} + 5x_{2i} + 7$. The factors considered for this experiment are dimension, outlier density, leverage (δ_L), outlying distance off the regression plane (δ_R) and the number of multiple point clouds. The full factorial 2^5 design and resulting measures of performance are displayed in Tables 5a (single cloud) and 5b (two clouds). A much more efficient 2_V^{5-1} design was initially run but many interesting factor combinations were missing.

The most notable feature from Tables 3.5a and 3.5b is the lack of detection capability for many of the methods now that leverage is added as a factor for $\delta_R = 3\sigma_e$ (the top half of both tables). These methods have not necessarily failed from one perspective because in most of these scenarios the outlying clouds do not breakdown the OLS parameters (note the moderate OLS false alarm rates). However, the practitioner still may want to identify these cases for reasons other than impact to estimation. The Sebert et al. and Rousseeuw and van Zomeren procedures do have detection power in these scenarios.

Table 3.5a. Design matrix with detection and false alarm probabilities for high-leverage (unusual in all k regressors) regression outliers forming a single cloud.

A n, k	B dens	C δ_L	D δ_R	E clds	HS93	HS97	K & S	P & Y	SM&R	R&vZ sim	R&vZ chi	GM	CE C&H	CE S&M	OLS
40, 2	10%	2σ	3σ	1	0.063 (.003)	0.070 (.002)	0.035 (.016)	0.585 (.027)	1.000 (.092)	0.490 (.053)	0.650 (.124)	0.253 (.071)	0.370 (.063)	0.510 (.067)	0.038 (.065)
60, 6	10%	2σ	3σ	1	0.021 (.002)	0.050 (.002)	0.000 (.028)	0.075 (.028)	1.000 (.057)	0.210 (.098)	0.210 (.221)	0.000 (.088)	0.000 (.092)	0.050 (.087)	0.000 (.060)
40, 2	20%	2σ	3σ	1	0.047 (.024)	0.050 (.013)	0.000 (.056)	0.195 (.038)	1.000 (.136)	0.050 (.184)	0.060 (.316)	0.000 (.144)	0.010 (.124)	0.020 (.123)	0.000 (.092)
60, 6	20%	2σ	3σ	1	0.011 (.031)	0.021 (.036)	0.000 (.064)	0.065 (.049)	1.000 (.102)	0.000 (.283)	0.000 (.412)	0.000 (.087)	0.000 (.121)	0.000 (.013)	0.000 (.059)
40, 2	10%	5σ	3σ	1	0.031 (.003)	0.010 (.002)	0.000 (.031)	0.385 (.032)	1.000 (.030)	1.000 (.049)	1.000 (.133)	0.000 (.063)	0.070 (.051)	0.060 (.061)	0.000 (.049)
60, 6	10%	5σ	3σ	1	0.026 (.003)	0.011 (.006)	0.000 (.031)	0.115 (.037)	1.000 (.019)	1.000 (.070)	1.000 (.174)	0.000 (.061)	0.000 (.071)	0.000 (.066)	0.000 (.044)
40, 2	20%	5σ	3σ	1	0.036 (.003)	0.021 (.027)	0.000 (.045)	0.045 (.057)	1.000 (.065)	0.970 (.116)	0.970 (.233)	0.000 (.063)	0.000 (.064)	0.000 (.072)	0.000 (.052)
60, 6	20%	5σ	3σ	1	0.015 (.011)	0.023 (.008)	0.000 (.082)	0.000 (.058)	1.000 (.047)	0.000 (.283)	0.000 (.410)	0.000 (.058)	0.000 (.081)	0.000 (.093)	0.000 (.040)
40, 2	10%	2σ	5σ	1	0.786 (.002)	0.620 (.019)	0.513 (.026)	0.990 (.024)	1.000 (.051)	0.930 (.044)	0.950 (.103)	0.961 (.074)	0.890 (.056)	0.902 (.057)	0.820 (.112)
60, 6	10%	2σ	5σ	1	0.320 (.006)	0.411 (.002)	0.008 (.029)	0.265 (.027)	1.000 (.045)	0.410 (.086)	0.440 (.189)	0.080 (.130)	0.250 (.111)	0.390 (.095)	0.080 (.094)
40, 2	20%	2σ	5σ	1	0.553 (.011)	0.530 (.008)	0.006 (.028)	0.470 (.035)	1.000 (.089)	0.330 (.167)	0.360 (.275)	0.071 (.258)	0.200 (.185)	0.390 (.151)	0.060 (.182)
60, 6	20%	2σ	5σ	1	0.267 (.038)	0.100 (.004)	0.000 (.041)	0.160 (.045)	1.000 (.083)	0.000 (.286)	0.000 (.416)	0.000 (.145)	0.000 (.175)	0.000 (.181)	0.000 (.109)
40, 2	10%	5σ	5σ	1	0.124 (.001)	0.180 (.002)	0.000 (.032)	0.580 (.022)	1.000 (.022)	1.000 (.044)	1.000 (.123)	0.000 (.098)	0.220 (.068)	0.240 (.084)	0.000 (.075)
60, 6	10%	5σ	5σ	1	0.020 (.002)	0.090 (.003)	0.000 (.027)	0.340 (.032)	1.000 (.017)	1.000 (.079)	1.000 (.188)	0.000 (.071)	0.010 (.081)	0.030 (.077)	0.000 (.055)
40, 2	20%	5σ	5σ	1	0.031 (.022)	0.160 (.019)	0.000 (.039)	0.250 (.059)	1.000 (.059)	0.970 (.116)	0.970 (.239)	0.000 (.098)	0.000 (.091)	0.010 (.100)	0.000 (.080)
60, 6	20%	5σ	5σ	1	0.095 (.013)	0.044 (.004)	0.000 (.078)	0.000 (.057)	1.000 (.045)	0.110 (.286)	0.110 (.406)	0.000 (.065)	0.000 (.097)	0.000 (.065)	0.000 (.047)

Table 3.5b. Design matrix with detection and false alarm probabilities for high leverage regression outliers (two clouds).

A n,k	B dens	C δ_L	D δ_R	E clds	HS93	HS97	K&S	P&Y	SM&R	R&vZ sim	R&vZ chi	GM	CE C&H	CE S&M	OLS
40, 2	10%	2σ	3σ	2	0.165 (.002)	0.165 (.003)	0.290 (.012)	0.753 (.030)	0.995 (.052)	0.685 (.039)	0.845 (.096)	0.620 (.071)	0.780 (.048)	0.895 (.052)	0.502 (.084)
60, 6	10%	2σ	3σ	2	0.000 (.001)	0.041 (.001)	0.007 (.019)	0.288 (.030)	1.000 (.034)	0.400 (.057)	0.570 (.143)	0.478 (.089)	0.358 (.072)	0.555 (.070)	0.198 (.073)
40, 2	20%	2σ	3σ	2	0.014 (.002)	0.024 (.000)	0.005 (.008)	0.408 (.041)	0.976 (.073)	0.360 (.106)	0.555 (.169)	0.485 (.145)	0.416 (.094)	0.628 (.083)	0.348 (.134)
60, 6	20%	2σ	3σ	2	0.025 (.000)	0.020 (.001)	0.000 (.017)	0.145 (.039)	1.000 (.043)	0.064 (.127)	0.271 (.219)	0.249 (.088)	0.040 (.099)	0.000 (.103)	0.000 (.081)
40, 2	10%	5σ	3σ	2	0.000 (.003)	0.001 (.000)	0.000 (.024)	0.513 (.027)	1.000 (.016)	1.000 (.032)	1.000 (.091)	0.365 (.063)	0.585 (.048)	0.630 (.054)	0.010 (.059)
60, 6	10%	5σ	3σ	2	0.000 (.001)	0.020 (.001)	0.000 (.016)	0.403 (.033)	1.000 (.011)	1.000 (.040)	1.000 (.115)	0.060 (.058)	0.180 (.060)	0.438 (.058)	0.000 (.048)
40, 2	20%	5σ	3σ	2	0.000 (.002)	0.025 (.012)	0.000 (.011)	0.373 (.034)	1.000 (.027)	0.970 (.025)	0.971 (.091)	0.101 (.063)	0.236 (.050)	0.489 (.055)	0.000 (.064)
60, 6	20%	5σ	3σ	2	0.000 (.001)	0.024 (.000)	0.000 (.024)	0.370 (.050)	1.000 (.017)	0.153 (.113)	0.363 (.202)	0.005 (.057)	0.000 (.063)	0.000 (.066)	0.000 (.044)
40, 2	10%	2σ	5σ	2	0.875 (.002)	0.855 (.001)	0.614 (.015)	0.993 (.025)	1.000 (.044)	0.975 (.037)	0.995 (.094)	0.980 (.074)	0.980 (.076)	0.995 (.045)	0.685 (.146)
60, 6	10%	2σ	5σ	2	0.168 (.016)	0.480 (.001)	0.034 (.018)	0.458 (.029)	1.000 (.030)	0.820 (.047)	0.873 (.121)	0.054 (.133)	0.825 (.067)	0.925 (.060)	0.465 (.122)
40, 2	20%	2σ	5σ	2	0.377 (.002)	0.504 (.001)	0.000 (.004)	0.540 (.037)	1.000 (.063)	0.860 (.103)	0.910 (.153)	0.534 (.255)	0.908 (.069)	0.950 (.060)	0.488 (.237)
60, 6	20%	2σ	5σ	2	0.087 (.001)	0.205 (.005)	0.000 (.011)	0.163 (.042)	1.000 (.047)	0.340 (.119)	0.423 (.201)	0.358 (.148)	0.250 (.130)	0.066 (.152)	0.031 (.145)
40, 2	10%	5σ	5σ	2	0.038 (.001)	0.210 (.001)	0.040 (.022)	0.505 (.027)	1.000 (.016)	1.000 (.030)	1.000 (.082)	0.470 (.098)	0.790 (.053)	0.825 (.055)	0.140 (.092)
60, 6	10%	5σ	5σ	2	0.000 (.001)	0.107 (.001)	0.000 (.018)	0.380 (.029)	1.000 (.011)	1.000 (.037)	1.000 (.112)	0.133 (.072)	0.407 (.066)	0.560 (.064)	0.000 (.062)
40, 2	20%	5σ	5σ	2	0.065 (.008)	0.155 (.005)	0.000 (.013)	0.335 (.033)	1.000 (.023)	0.990 (.022)	0.985 (.079)	0.123 (.099)	0.491 (.072)	0.615 (.068)	0.000 (.102)
60, 6	20%	5σ	5σ	2	0.027 (.009)	0.055 (.004)	0.000 (.019)	0.330 (.046)	1.000 (.018)	0.130 (.112)	0.303 (.204)	0.011 (.067)	0.000 (.077)	0.000 (.076)	0.000 (.055)
Average probabilities					0.134 (.007)	0.165 (.006)	0.049 (.028)	0.359 (.037)	0.999 (.046)	0.601 (.103)	0.650 (.192)	0.200 (.099)	0.290 (.084)	0.349 (.079)	0.121 (.086)
Significant effects detection capability					A, C, D,CD	A, B, C, D, AD, CD	A, B, C, AB, AC, BC	A, B, C,D,E AB, AC	none	A, B, C, AB, CE	A, B, C, AB, AE, CE	A, B, C, E, AB	A, B, C, D, E	A, B, C, D, E	A, C
Significant effects false alarms					B, E, BE	D, E, DE	A, B, C, E, BE, CE	A, B, C, E, BE, CE	A, B, C, E	B, E, BE	A, B, E, AE, BE	B, C, D, AB, BC	A, B, C, D, E, BC	B	C

For the direct methods in all scenarios, the Sebert et al. procedure has virtually perfect detection capability and reasonable false alarm rates. This success is attributed to a favorable clustering condition for the outliers from unusually high predicted response values coupled with near zero standardized least squares residuals because of the leverage. The expected response value for the clean observations is $E(y) = 5k * 7.5$ and the expected response for the outliers is $5k * (7.5 + 4\delta_L) + .125 + \delta_R$. We investigate the algorithm's performance if the predicted response values are not unusual (Y-space inliers) in the next section, 3.4.2.2. The other direct methods do not fare as well. Both the Hadi and Simonoff and Swallow and Kianifard methods have little detection capability in almost all scenarios because they sequentially add an observation to the clean basis as a function of the smallest OLS residual. Clearly, these high-leverage outliers can have very small OLS residual values and are often masked. We note again the unusually low false alarm probabilities for both Hadi and Simonoff procedures and investigate the possibility of relaxing the cutoff values from the t distribution in Section 3.4.2.3. The Pena and Yohai algorithm does have some moderate detection capability for these high-leverage regression outlier scenarios.

The Rousseeuw and van Zomeren methods successfully detect the outlying clouds in exterior X-space but are troubled by high false alarm rates; particularly for the single cloud scenarios in Table 3.5a. The simulated cutoff values provide slightly less outlier detection capability but significantly lower false alarm rates than the original proposal.

For the indirect methods with regression estimators, the generalized M-estimate has poor detection capability because of the breakdown of the OLS initial estimate and the

hat diagonal leverage component. The compound estimators have reasonable performance in the high residual distance scenarios in Table 3.5b, apart from the high dimension, high contamination scenarios. The false alarm probability moderately exceeds the nominal 5% rate for both compound estimators in these scenarios. The Simpson and Montgomery estimator slightly outperforms the Coakley-Hettmansperger estimator.

3.4.2.2 Regression Outliers in Clouds that are Unusual in X-Space in All k Regressors but the Outlier Responses are not Unusual in Y-Space

This study investigates the effect of changing geometry in X-space such that the outlying cloud will not have an unusual response value with respect to the responses for the clean observations. The values for the ith regressor variable for the outliers are now generated as $\bar{x}_{i,clean} + 4\delta_L + \varepsilon_{ij}^*$ for $i = 1, 3, 5$ and $\bar{x}_{i,clean} - 4\delta_L + \varepsilon_{ij}^*$ for $i = 2, 4, 6$. This scheme effectively equalizes the expected response values for the clean and outlying cases. The four scenarios in Table 3.6 are randomly selected from those in Section 3.4.2.1 with the regressor variable values for the outliers generated as described above.

The results in Table 3.6 for nearly all techniques are within a few percentage points for both detection capability and false alarm probabilities. The 1997 Hadi and Simonoff procedure has significantly lower detection capability than most other procedures for the first scenario and significantly higher detection capability in the last scenario. In contrast to near perfect performance in the previous experiment, the Sebert et al. procedure fails in

Table 3.6. Design matrix with detection capability and false alarm probabilities (in parentheses) for high-leverage regression outliers when the response variable is not unusual in Y-space for the outlying observations.

A n, k	B dens	C δ_L	D δ_R	E clds	HS93	HS97	K & S	P & Y	SM&R	R&vZ sim	R&vZ chi	GM	CE C&H	CE S&M	OLS
40, 2	10%	2σ	5σ	2	0.936 (.005)	0.610 (.000)	0.605 (.018)	0.998 (.028)	0.525 (.139)	0.940 (.028)	0.990 (.084)	0.980 (.074)	0.990 (.047)	1.000 (.043)	0.711 (.156)
60, 6	10%	2σ	5σ	1	0.423 (.002)	0.521 (.002)	0.011 (.033)	0.260 (.025)	0.011 (.205)	0.440 (.078)	0.453 (.174)	0.051 (.142)	0.250 (.123)	0.450 (.094)	0.010 (.103)
40, 2	20%	5σ	5σ	1	0.126 (.008)	0.211 (.019)	0.000 (.057)	0.335 (.065)	0.057 (.203)	0.962 (.122)	0.962 (.246)	0.000 (.100)	0.000 (.101)	0.000 (.104)	0.000 (.087)
60, 6	20%	5σ	5σ	2	0.505 (.008)	0.516 (.007)	0.004 (.020)	0.295 (.046)	0.021 (.184)	0.100 (.117)	0.290 (.206)	0.011 (.068)	0.000 (.074)	0.000 (.056)	0.000 (.078)
Average probabilities					0.498 (.006)	0.465 (.007)	0.155 (.032)	0.472 (.041)	0.154 (.183)	0.611 (.086)	0.674 (.178)	0.261 (.096)	0.310 (.086)	0.363 (.074)	0.180 (.106)

these scenarios. Not only is detection capability low, but also the false alarm probability is high.

3.4.2.3 Outliers are Unusual in X-space in a Subset of Regressor Variables and Larger Residual Magnitude (δ_R) Factor Settings

This study investigates the power and false alarm rates for the procedures when the factor settings for residual magnitude are changed from $\delta_R = 3\sigma_e$ for the low level and $5\sigma_e$ for the high level to $5\sigma_e$ and $10\sigma_e$ respectively. The number of clouds is set at one because this has proven to be the more challenging configuration for these procedures. The number of unusual regressor variables out of k for the outliers is introduced as a factor with the low level as 1 and the high level as 2 for $k = 2$ and 3 for $k = 6$. We believe this to be a more likely scenario to encounter in practice as opposed to finding cases that are outlying in all k variables. Additionally, the regressor variables alternate in sign as described in Section 3.4.2.2 to guard against unusually large response values for the planted outliers. The experiment design in Table 3.7 is a 2_V^{5-1}; the two-factor interactions are not aliased with main effects or other two-factor interactions.

These scenarios are important to detect because of significant swamping from the OLS fit as evidenced by the high false alarm rates in Table 3.7. The shaded scenarios indicate voids where all procedures fail to detect the outlying cloud in high dimension, high contamination. Noteworthy results for the direct procedures across all scenarios are

Table 3.7. Half-fraction design matrix with detection and false alarm probabilities (in parenthesis) for large regression outlying distance. The outliers are not necessarily outlying in all regressor variables.

A n,k	B dens	C δ_L	D δ_R	E k_o	HS93 (0.05)	HS93 (0.20)	HS97 (0.05)	HS97 (0.20)	K&S	P&Y	SM&R	R&vZ sim	R&vZ chi	GM	CE C&H	CE S&M	OLS
40, 2	10%	2σ	5σ	2	0.881 (.002)	0.950 (.013)	0.750 (.001)	0.692 (.012)	0.490 (.019)	0.995 (.022)	0.350 (.175)	0.895 (.014)	0.965 (.059)	0.985 (.068)	0.960 (.053)	0.990 (.043)	0.888 (.123)
60, 6	10%	2σ	5σ	1	0.910 (.002)	0.920 (.009)	0.614 (.001)	0.557 (.019)	0.860 (.010)	0.890 (.027)	0.893 (.122)	0.930 (.047)	0.990 (.017)	1.000 (.068)	1.000 (.055)	1.000 (.068)	1.000 (.111)
40, 2	20%	2σ	5σ	1	0.874 (.006)	0.892 (.040)	0.753 (.012)	0.756 (.033)	0.093 (.018)	0.835 (.026)	0.853 (.135)	0.821 (.042)	0.885 (.081)	0.864 (.149)	0.835 (.095)	0.935 (.066)	0.828 (.215)
60, 6	20%	2σ	5σ	3	0.427 (.017)	0.454 (.084)	0.091 (.005)	0.094 (.157)	0.000 (.047)	0.040 (.039)	0.103 (.209)	0.060 (.228)	0.060 (.356)	0.000 (.232)	0.000 (.235)	0.000 (.273)	0.000 (.163)
40, 2	10%	5σ	5σ	1	0.310 (.004)	0.450 (.022)	0.331 (.003)	0.531 (.008)	0.027 (.025)	0.663 (.019)	0.995 (.044)	0.955 (.024)	0.965 (.086)	0.190 (.113)	0.660 (.066)	0.640 (.068)	0.024 (.103)
60, 6	10%	5σ	5σ	3	0.155 (.002)	0.255 (.023)	0.187 (.028)	0.274 (.011)	0.000 (.030)	0.195 (.030)	0.508 (.153)	1.000 (.062)	1.000 (.167)	0.000 (.087)	0.060 (.088)	0.060 (.087)	0.000 (.065)
40, 2	20%	5σ	5σ	2	0.074 (.014)	0.082 (.173)	0.214 (.019)	0.230 (.138)	0.000 (.060)	0.150 (.038)	0.013 (.199)	1.000 (.073)	1.000 (.187)	0.000 (.097)	0.000 (.086)	0.000 (.101)	0.000 (.086)
60, 6	20%	5σ	5σ	1	0.275 (.024)	0.305 (.129)	0.263 (.005)	0.269 (.109)	0.000 (.060)	0.067 (.043)	0.630 (.194)	0.020 (.234)	0.025 (.362)	0.000 (.149)	0.000 (.172)	0.000 (.183)	0.000 (.110)
40, 2	10%	2σ	10σ	1	0.993 (.002)	0.990 (.015)	0.962 (.001)	0.988 (.012)	1.000 (.004)	1.000 (.017)	1.000 (.025)	1.000 (.014)	1.000 (.061)	1.000 (.061)	1.000 (.044)	1.000 (.043)	1.000 (.283)
60, 6	10%	2σ	10σ	3	0.710 (.002)	0.776 (.006)	0.942 (.003)	0.937 (.003)	0.925 (.041)	0.995 (.021)	0.755 (.156)	1.000 (.047)	1.000 (.109)	1.000 (.104)	1.000 (.059)	1.000 (.049)	1.000 (.299)
40, 2	20%	2σ	10σ	2	0.754 (.006)	0.751 (.072)	0.853 (.002)	0.860 (.033)	0.128 (.059)	0.855 (.025)	0.123 (.203)	0.990 (.027)	0.990 (.057)	0.629 (.393)	0.960 (.070)	1.000 (.041)	0.959 (.448)
60, 6	20%	2σ	10σ	1	0.809 (.001)	0.837 (.019)	0.204 (.005)	0.204 (.085)	0.655 (.083)	0.812 (.034)	0.640 (.203)	1.000 (.019)	0.990 (.167)	1.000 (.193)	1.000 (.057)	1.000 (.043)	1.000 (.469)
40, 2	10%	5σ	10σ	2	0.448 (.005)	0.440 (.021)	0.626 (.002)	0.675 (.010)	0.040 (.033)	0.743 (.017)	0.030 (.170)	1.000 (.015)	1.000 (.063)	0.123 (.214)	0.900 (.059)	0.940 (.051)	0.078 (.194)
60, 6	10%	5σ	10σ	1	0.615 (.002)	0.611 (.014)	0.919 (.002)	0.874 (.008)	0.208 (.051)	0.748 (.023)	0.950 (.121)	0.990 (.039)	0.990 (.104)	0.760 (.208)	0.980 (.057)	0.970 (.052)	0.773 (.248)
40, 2	20%	5σ	10σ	1	0.380 (.019)	0.380 (.111)	0.774 (.011)	0.701 (.070)	0.003 (.038)	0.280 (.029)	1.000 (.055)	0.950 (.040)	0.950 (.089)	0.010 (.354)	0.625 (.135)	0.790 (.088)	0.088 (.308)
60, 6	20%	5σ	10σ	3	0.154 (.022)	0.150 (.178)	0.701 (.007)	0.510 (.105)	0.000 (.067)	0.153 (.047)	0.600 (.193)	0.011 (.237)	0.010 (.369)	0.000 (.185)	0.000 (.198)	0.000 (.212)	0.000 (.162)
Average probabilities					0.548 (.008)	0.578 (.058)	0.574 (.007)	0.572 (.051)	0.277 (.040)	0.589 (.029)	0.590 (.147)	0.789 (.073)	0.801 (.146)	0.473 (.167)	0.624 (.096)	0.645 (.092)	0.477 (.212)
Significant effects detection capability					C	C	D	D	B, C	B, C	B, E, AE	A, B, AB	A, B, AB	B, C, E	B, C, D	B, C, D	C
Significant effects false alarms					A, B, AB	B	none	B	B	B, C	A, B, E, AE	A, B	A, B	B, D	B	A, B, AB	C

the general failure of the Swallow and Kianifard recursive residuals procedure, the high false alarm rate and limited detection capability of the Sebert et al. clustering procedure, and the improved performance (although somewhat limited in detection capability in high-density scenarios) of the Pena and Yohai influence matrix procedure. The original Hadi and Simonoff forward selection procedure performs better than the improved version in low-leverage scenarios and the opposite is true for exterior X-space. Relaxing α to 0.20 gains very little in detection capability for both versions of the Hadi and Simonoff procedure but carries the risk of excessive false alarms in high-dimension scenarios.

Except for the shaded scenarios, the Rousseeuw and van Zomeren procedures have near perfect detection capability. Both suffer from high false alarms, although the simulated critical value procedure has slightly lower false alarm rates. The robust regression estimators have consistent results with the previous sections: 1) the failure of the generalized M-estimator in high-leverage scenarios, 2) the Simpson & Montgomery estimator slightly outperforms the Coakley & Hettmansperger estimator and 3) the compound estimators have difficulty with some exterior X-space scenarios.

3.4.3 Interior and Exterior X-Space Outliers

This study evaluates the performance of the procedures when large magnitude outliers in both δ_L and δ_R are present that could mask the lower magnitude outliers. The first four scenarios have a single cloud with 10% of the observations outlying at high-leverage ($\delta_L = 5\sigma_x$) and significantly off the regression plane ($\delta_R = 10\sigma_e$). Another 10% of

the observations are randomly scattered regression outliers at the interior of X-space at a magnitude of $\delta_R = 4\sigma_e$. The last four scenarios place the interior outliers in a cloud approximately at the centroid of X-space, $\delta_R = 4\sigma_e$.

The original Rousseeuw and van Zomeren procedure has the best detection probabilities in Table 3.8 and also has false alarm rates slightly above the nominal 5% level. The next best performing methods are the compound estimators, also with moderately high false alarm probabilities. The Sebert et al. procedure is the only direct procedure with any significant detection capability. All procedures identify the outliers better if the number of outlying regressor variables is one because of the decrease in influence exerted from the high-leverage points.

3.5 Procedure Summary and Recommendations

The most interesting performance characteristics of the various procedures have been noted in the results for each study. This section provides a summary of those results by procedure and discusses the powerful and vulnerable areas of performance.

3.5.1 Performance Summary of Direct Procedures

Hadi and Simonoff. Both versions are powerful in all of the experiments in Section 3.4.1 when the regression outliers are in the interior of X-space. The most notable feature in these scenarios is the very low false alarm probability. This prompted an

Table 3.8. Design matrix with detection and false alarm probabilities (in parenthesis) for large magnitude high leverage outliers and smaller magnitude low leverage outliers.

A n,k	B var	C place	HS93	HS97	K & S	P & Y	Sebert	R & vZ Sim	R & vZ Pub	GM	CE C & H	CE S & M	OLS
40, 2	2	random	0.319 (.002)	0.454 (.002)	0.184 (.024)	0.461 (.015)	0.684 (.011)	0.835 (.031)	0.961 (.069)	0.438 (.223)	0.853 (.078)	0.903 (.066)	0.473 (.208)
60, 6	3	random	0.133 (.003)	0.256 (.001)	0.153 (.011)	0.112 (.012)	0.771 (.038)	0.763 (.031)	0.926 (.085)	0.443 (.186)	0.708 (.099)	0.747 (.093)	0.443 (.178)
40, 2	1	random	0.443 (.004)	0.465 (.001)	0.181 (.019)	0.426 (.013)	0.731 (.065)	0.813 (.024)	0.969 (.058)	0.729 (.235)	0.965 (.056)	0.991 (.051)	0.776 (.281)
60, 6	1	random	0.283 (.001)	0.439 (.001)	0.063 (.009)	0.127 (.011)	0.513 (.113)	0.750 (.026)	0.928 (.061)	0.582 (.261)	0.981 (.059)	0.995 (.052)	0.697 (.269)
40, 2	2	cloud	0.344 (.001)	0.137 (.001)	0.104 (.013)	0.350 (.012)	0.795 (.010)	0.860 (.048)	0.990 (.096)	0.515 (.231)	0.875 (.073)	0.910 (.083)	0.525 (.214)
60, 6	3	cloud	0.065 (.000)	0.248 (.002)	0.140 (.014)	0.097 (.018)	0.782 (.038)	0.763 (.038)	0.938 (.089)	0.492 (.175)	0.773 (.089)	0.837 (.080)	0.493 (.168)
40, 2	1	cloud	0.351 (.001)	0.459 (.000)	0.085 (.022)	0.373 (.014)	0.741 (.067)	0.858 (.027)	0.988 (.062)	0.751 (.246)	0.995 (.048)	1.000 (.056)	0.773 (.289)
60, 6	1	cloud	0.308 (.009)	0.372 (.002)	0.015 (.008)	0.077 (.013)	0.539 (.135)	0.780 (.024)	0.944 (.058)	0.641 (.266)	0.992 (.053)	0.988 (.051)	0.722 (.278)
Average probabilities			-0.281 (.003)	0.354 (.001)	0.116 (.015)	0.253 (.014)	0.695 (.060)	0.803 (.031)	0.956 (.072)	0.574 (.228)	0.893 (.069)	0.921 (.067)	0.613 (.236)
Significant effects Detection capability			A, B	B		A	B, AB	A	A	B	B	B	B
Significant effects false alarms					A		A, B	B	B		B	B	B

increase of $\alpha = 0.05$ to $\alpha = 0.20$ to compute the cutoff value from the t distribution using the Bonferroni approach. Dramatic increases in detection probabilities from this enhancement are realized in our selected scenarios accompanied by false alarm probabilities well below the nominal 5% rate. Detection capability moderately declines in the high-dimension, high-density scenarios. The original 1993 version outperforms (especially at $\alpha = 0.20$) or is equivalent to the robust 1997 version in virtually all of the experiments in Section 3.4.1 because an initial basis of robust distances does little when the outliers are not leverage points.

Overall performance noticeably degrades for these two algorithms as leverage is added as a factor. This can be attributed to the loss of signal from the OLS studentized residuals and scaled prediction errors. Increasing α to 0.20 does not increase detection capability and may swamp too many clean observations in the high-leverage scenarios of Section 3.4.2. Also in these scenarios, indirect methods significantly outperform both versions of the algorithm. Detection capability is increased to reliable levels if the outlying distance off the regression plane (δ_R) is sufficiently large relative to the leverage δ_L. In the higher leverage scenarios, the robust 1997 algorithm outperforms the original algorithm.

Swallow and Kianifard. This algorithm, based on recursive residuals from a least squares fit using a robust scale estimate, reliably detects regression outliers in the interior of X-space at $\delta_R = 4\sigma_e$ and beyond. High-dimension, high-density scenarios affect detection capability. The detection capability of this algorithm is also highly sensitive to leverage and it lags behind the other procedures for the regression outliers unusual in X-

space studies. Despite the lack of power in these scenarios, the false alarm rate rarely exceeds the nominal 5% rate anywhere.

Pena and Yohai. Of the direct procedures, the Pena and Yohai algorithm with the eigenanalysis of the influence matrix may be the most versatile. Although it does not detect regression outliers in interior X-space until $5\text{-}7\sigma_e$, it does detect the high-leverage regression outliers reasonably well. Also, the procedure rarely swamps clean observations. The scenarios presented in this paper are challenging; however, in practice, the scenarios of interest may have magnitudes of the outlying distances (δ_L and δ_R) large enough to effectively use this procedure.

Sebert et al. The clustering algorithm of the least squares standardized predicted and residual values is often the only procedure with any detection capability at all. For this method to be successful, a signal has to come from one or both of these quantities. In the scenarios of Section 3.4.1, the signal comes from the standardized residual values only and the procedure is competitive here with the others in detection capability but has false alarms rates often 2 to 3 times the nominal level. As the outlying distance off the regression surface, δ_R, increases the false alarm rate decreases and the detection capability increases. The procedure works especially well for the exterior X-space outliers in Section 3.4.2 if the predicted response values for the outliers are unusual with respect to the clean response values. The standardized least squares residuals of the outliers are often close to 0 in the high-leverage scenarios so the outlying clusters must form in the algorithm from unusual predicted values. In Section 3.4.2.2 we show that the method is

vulnerable in high-leverage scenarios if the outliers are not unusual in predicted value. If

the assumption that the data will be unusual in at least one of the two measures is met, this

is a very powerful, yet easily implemented algorithm.

3.5.2 Performance Summary of Indirect Procedures

High breakdown estimators. Both LMS and LTS detect regression outliers in the

interior of X-space well if the outlying distance δ_R is at least $4\sigma_e$. Both estimators also

have low false alarm rates across all scenarios when δ_R is at least $4\sigma_e$, unlike many of the

other regression estimators evaluated in Section 3.4.2. One notable exception is the high

false alarms rates in the scenarios of Table 3.4 in high dimension with 30% outlier density.

Overall, LMS detects the outliers in high dimension slightly better than LTS. Detection

capability decreases for both estimators as the outliers become more remote in X-space.

Although not tested separately for regression outliers remote in X-space, theory and our

pilot studies showed LMS and LTS to have significant masking and swamping problems in

high-leverage cases.

The Rousseeuw and van Zomeren technique that combines the robust distances

from the MVE with the scaled residuals from an LMS fit is one of the better performing

techniques for exterior X-space regression outliers. Again, we prefer the simulated cutoff

values to protect against swamping too many observations. The weak areas are limited to

high-dimension, high-density.

M and MM estimators. These estimators are only evaluated in the regression

outliers in the interior of X-space of Section 3.4.1 because they are known to fail in the

high-leverage experiments of Sections 3.4.2. The results from the first three experiments in Section 3.4.1 indicate that the detection capability and false alarm rate of the *MM* estimator is only slightly preferred over the *M* estimator. For these scenarios, both estimators have excellent detection power but have moderate false alarm problems. The high outlying magnitude, high-density runs in Section 3.4.1.4 demonstrate the superiority of the *MM* estimator. Despite the comparable detection probabilities, the *M*-estimator breaks down and suffers from severe false alarm rates in these scenarios.

GM and compound estimators. The standard generalized-*M* bounded-influence estimator is plagued by a higher false alarm rate and lower detection capability than the compound estimators in the exterior X-space regression outlier experiments in Section 3.4.2. This effect is most evident in the high-density scenarios where both the OLS initial estimate and the hat diagonal component for leverage breakdown for this estimator. The compound estimators do a decent job identifying the high-leverage multiple outliers. Both the Simpson and Montgomery and Coakley and Hettmansperger estimators have similar detection capability and moderately high false alarm probabilities in many scenarios. In several exterior X-space scenarios, only the compound estimators and the Rousseeuw and van Zomeren procedure successfully detect these outliers. From Table 5a, both have little detection capability with moderate leverage despite relatively large residual magnitudes. This presents a research opportunity that will be explored in Chapter 4.

3.5.3 Summary of Results

The simulation experiments in this paper validate many of expected performance characteristics of the multiple outlier detection methods. As a general rule, the detection methods perform better in lower dimension, lower outlier density, smaller outlying leverage distance, larger outlying residual distance, and larger number of multiple point clouds. However, we show scenarios where this is not the case for all methods and all factors. Some factors are shown to be either not significant or behave opposite to the general rule. The most important findings suggest that limited studies in low dimension of a proposed procedure are not sufficient to speculate on its performance in higher dimension—especially if the percentage outliers is large. From the interior X-space studies of Section 3.4.1, the high-breakdown methods perform well. *MM* performs the best overall. The 1993 version of the Hadi and Simonoff algorithm can be recommended if the residual outlying distance is large. For the exterior X-space studies in Section 3.4.2, the compound estimators and the robust distance with high-breakdown estimator procedures perform the best. The Simpson and Montgomery estimator and the Rousseeuw and van Zomeren method with simulated cutoff values show the best results in our studies.

Chapter 4

An Improved Robust Regression Compound Estimator

4.1 Introduction

Barnett and Lewis (1994) define outliers as observations that appear inconsistent with the remainder of the data set. In the linear regression model, we consider three classes of outliers: 1) residual or regression outliers, whose response values differ significantly from those expected from the fit with uncontaminated data, 2) leverage outliers, whose regressor variable values are extreme in X-space and 3) observations that are both residual and leverage outliers. A single outlier in an ordinary least squares (OLS) regression model could be placed to alter the parameter estimates such that the fit to the remaining $n - 1$ data points is poor. Fortunately, many standard least squares regression diagnostic quantities and plots can reliably identify a single or a few of these three types of outliers. One modeling approach in the presence of outliers is to remove the discordant observations from the model and fit the remaining observations. Robust regression estimators offer an alternative between removing the outliers and including them in the model by weighting each observation as a function of "outlyingness".

Numerous robust regression estimators exist. It is generally accepted that no single estimator optimally protects against all outlier scenarios likely to be encountered in practice. The properties of a good robust regression estimator are 1) high-breakdown, 2) efficient and 3) bounded-influence. High-breakdown estimators can fit a model to the bulk of the data even if a large percentage of outliers (as much as 50% for some

estimators) are present. Least squares has a breakdown of 0% because a single outlying observation can be placed in a data set that makes the parameter estimates and inferences for the remaining $n - 1$ observations meaningless. An efficient estimator provides parameter estimates close to those from an OLS (the best linear unbiased estimator) fit in an uncontaminated sample with NID error terms. Bounded-influence estimators protect the regression surface from being pulled toward extreme observations in X-space. OLS estimators do not have bounded-influence and the more extreme the outlier is in X-space, the greater the impact it has on the parameter estimates.

Theoretical and simulation results in the literature show that many robust regression estimators are vulnerable with respect to at least one of the three desirable properties. For example, the common high-breakdown estimators suffer from inefficiency and unbounded-influence while many efficient techniques are not high-breakdown nor bounded-influence. Multi-staged techniques have been proposed to combine several of the properties into a single estimator. There exist multi-staged compound and generalized M-estimators (GM) with all three properties that can accommodate data sets with all three classes of outliers.

Compound and GM estimators downweight outlying observations by minimizing a function of the residuals rather than the sum of the squared residuals (OLS). Parameter estimates are obtained by solving a system of nonlinear normal equations. The normal equations incorporate a leverage measure to accommodate high-leverage points and a robust measure of scale. An iteration scheme to solve the normal equations requires good initial parameter estimates; these are often from a high-breakdown estimator. A

compound estimator uses only a single iteration to solve the nonlinear normal equations to preserve the high-breakdown property (Simpson, Ruppert and Carroll, 1992, and Yohai, 1997). *GM* estimators use a fully iterated scheme and have a breakdown of $1/p$.

There have been some empirical performance studies of robust regression estimators (Simpson and Montgomery, 1998a, 1998b, Wilcox, 1997, and Meintanis and Donatas, 1997). The best performing estimators with respect to breakdown, bounded-influence, efficiency and robustness to outlier scenarios appear to be the compound estimators of Coakley and Hettmansperger (1993) (C&H) and Simpson and Montgomery (1998a) (S&M). This chapter proposes several compound estimators with alternative high-breakdown initial estimators and measures of leverage and recommends a single method.

Section 4.2 explains *GM* and compound estimators. An example in Section 4.3 exposes some vulnerabilities in the measures of leverage and initial estimates for published compound estimators. Section 4.4 is an extensive Monte Carlo performance study of some measures of leverage. Section 4.5 incorporates the best performing measure of leverage from 4.4 and develops the need for a better initial estimator. Section 4.6 tests several common and proposed initial high-breakdown estimators. We propose a new compound estimator in Section 4.7, conduct performance studies in Section 4.8, and summarize results in Section 4.9.

4.2 Compound Estimators in Linear Regression

The standard linear regression model is $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where $\mathbf{y}$ is the observed response vector of dimension n, the number of observations; $\mathbf{X}$ is the observed $n \times p$ matrix of regressor variables with intercept; $\boldsymbol{\beta}$ is the vector of regression parameters, and $\boldsymbol{\varepsilon}$ is the column vector of n random errors assumed to have mean $\mathbf{0}$ and covariance matrix $\sigma^2 \mathbf{I}$. *GM*-estimators were offered as improvements to their predecessor *M*-estimates (maximum likelihood) to protect against high-leverage outliers. Rather than minimize the sum of squared errors as the objective, the *M*-estimate minimizes a function ρ of the errors. The *M*-estimate objective is $\min\limits_{\beta} \sum\limits_{i=1}^{n} \rho\left(\dfrac{e_i}{s}\right) = \min\limits_{\beta} \sum\limits_{i=1}^{n} \rho\left(\dfrac{y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{s}\right)$ where s is an estimate of scale often formed from a linear combination of the residuals. The system of normal equations to solve this minimization problem is found by taking partial derivatives with respect to β and setting them equal to 0, yielding $\sum\limits_{i=1}^{n} \psi\left(\dfrac{y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{s}\right)\mathbf{x}_i = \mathbf{0}$ where ψ is the derivative of ρ.

The choice of the ψ-function is based on the preference of how much weight to assign outliers (see e.g. Montgomery and Peck, 1992). A monotone ψ-function does not weight large outliers as much as least squares (e.g. a 10σ outlier would receive the same weight as a 3σ outlier). A redescending ψ-function increases the weight assigned to an outlier until a specified distance (e.g. 3σ) and then decreases the weight to 0 as the outlying distance gets larger.

The *GM*-estimator bounds the influence in leverage by weighting the *M*-estimate system of normal equations by a measure of leverage. The *GM* system of normal equations is $\sum_{i=1}^{n} \pi_i \psi \left(\frac{y_i - \mathbf{x}_i'\hat{\beta}}{s\pi_i} \right) \mathbf{x}_i = \mathbf{0}$ where π_i is a measure of remoteness in X-space.

When the π-weights are located both inside and outside the argument of the ψ-function, the *GM* objective function is Schweppe (Handsin et al., 1975). If the π-weights are not inside the argument, then the *GM* objective function is Mallows (Mallows, 1975). In practice, the distinction between the two objective functions is that Mallows will downweight high-leverage points independently of the residual value while Schweppe will not downweight if the response value conforms to the regression surface. Thus, Mallows does not incorporate "good outliers" in the parameter estimates. Several approaches to forming the π-weights use a distance measure from either the hat diagonal ($h_{ii} = \mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i'$), *M*-estimates of covariance, the minimum volume ellipsoid (MVE) or the minimum covariance determinant (MCD). These methods and some proposed methods are described in Section 4.4.

A numerical optimization scheme is required to solve the *GM* system of nonlinear normal equations. The two most common approaches are Newton's method and iteratively reweighted least squares (IRLS). Both approaches require initial parameter estimates for β. Most initial estimators are selected to provide decent parameter estimates in the presence of a large percentage (as much as 50% in some cases) of outliers. The popular choices for these high-breakdown initial estimates are the least median of squares (Rousseeuw, 1984) (LMS), least trimmed sum of squares (Rousseeuw,

1985) (LTS), and *S*-estimation (Rousseeuw and Yohai, 1984). The final parameter estimates from the optimization routine can come from a fully iterated solution (*GM*-estimator) or only a single iteration (compound estimator). The single iteration method preserves the breakdown of the initial estimator.

Simpson and Montgomery (1998b) test several *GM* and compound estimators and find that the Simpson and Montgomery (1998a) estimator and the Coakley and Hettmansperger (1993) estimator have good overall performance. The S&M estimator uses an *S*-estimate that minimizes the dispersion of the residuals for both the initial parameter estimates and the measure of scale. Other components are modified *M*-estimates of covariance distances to form the π-weights, a Schweppe *GM* objective function, a redescending Tukey ψ-function and a one-step reweighted least squares convergence criteria. The C&H estimator uses an LTS initial estimate, an LMS estimate of scale (the initial estimate's scaled median residual), robust distances from an MVE estimator for the π-weight component, a monotone Huber ψ-function and solves the normal equations with a single iteration of a Newton algorithm.

4.3 Compound Estimator Example

Consider creating a regression data set of $n = 60$ observations and $k = 6$ regressor variables with 12 high-leverage residual outliers. The outliers are remote in X-space because the values for their first two of six regressors are 5 standard deviations above the mean of the clean regressor variable values. The response values for these outliers are 10 standard deviations away from the regression surface defined by the fit from the clean 48

cases. Table 4.1 describes how the regressor and response variables are generated for both the clean and outlying cases. The last 12 cases in the data set (shown in Appendix B) are the planted outliers.

Table 4.1. Generating distributions for example 4.1. $\beta_0 = 0$ and $\beta_j = 3$ for $j = 1$ to 6.

Case	x_{ij} $j = 1, 2$	x_{ij} $j = 3, 4, 5, 6$	y_i	ε_i
$1 - 48$	NID $(7.5, 4^2)$	NID $(7.5, 4^2)$	$\mathbf{x}_i \beta + \varepsilon_i$	NID $(0, 1)$
$49 - 55$	$27.5 + \text{UNIF}(0,1)$	NID $(7.5, 4^2)$	$\mathbf{x}_i \beta + 10$	0
$55 - 60$	$29.5 + \text{UNIF}(0,1)$	NID $(7.5, 4^2)$	$\mathbf{x}_i \beta + 10$	0

Because there is a large percentage of high-leverage points, a *GM* or compound estimator is likely to be our best choice to accommodate the outliers. We choose the S&M and C&H estimators. Both estimators erroneously fit the 12 outliers (residuals near 0) and assign weights of nearly 100% to these observations. By chasing the outliers, the fit for the 48 clean cases is degraded. Many clean cases (8 for S&M and 7 for C&H) now have large residuals that a researcher could erroneously label as outliers. The mean squared error (MS_E) for the 48 clean cases using the S&M and C&H parameter estimates is more than three times the MS_E obtained by a least squares fit to the clean data. Another problem is that the π-weights for these high-leverage observations are not unusual. If the contamination is reduced in this example to 10% from 20% or if the leverage distance is reduced to 2 standard deviations above the mean from 5, for example, the outlying observations are correctly downweighted for both S&M and C&H.

From the performance studies in Chapter 3, the S&M and C&H estimators are successful across a variety of outlier scenarios, but are vulnerable (as are all techniques) in the high-leverage, high-density, high-dimension scenarios of this example. A possible

solution to the problem is to find better estimates of leverage because the π-weights are not providing any indication of unusual geometry in X-space.

4.4 A Performance Study for Measures of Leverage

This section describes several measures of leverage from the literature that can be used to form π-weights. Monte Carlo simulations using factorial designs with factors thought to impact performance provide a comprehensive test of each procedure across numerous X-space conditions. The goal is to possibly improve an existing *GM* or compound estimator by finding a technique or a combination of techniques that performs well in most scenarios likely to be encountered in practice.

The standard measure of leverage in OLS is the hat diagonal element. This quantity is often used as a measure of "outlyingness" in X-space and is extensively used in influence diagnostic quantities. Remote observations in X-space may exert enough influence on the least squares estimates to make them quite different from those obtained with only the observations in the interior of X-space. Some *GM* estimators (e.g. Walker, 1984) incorporate the hat diagonal measures of remoteness in X-space to accommodate these outlying observations. The hat diagonal measure may not provide an adequate leverage measure when there is even a moderate number of outliers in X-space present because the covariance matrix estimate is significantly influenced or "pulled" toward the outliers. For the data set of Example 4.1 using only the first 49 observations, the outlier (observation 49) has a hat diagonal value of 0.60 which exceeds the usual cutoff $(3p/n)$ of 0.42. However, for the full data set, the hat diagonals are not at all unusual for the 12

outliers because the outliers have significantly altered the covariance matrix. Therefore, the hat diagonal has broken down in the presence of multiple outliers and does not provide a reliable measure of leverage.

High-breakdown measures of leverage have been proposed that use a robust measure of the mean and covariance matrix in the standard Mahalanobis distance computation, $D_i^2 = (\mathbf{x}_i - \overline{\mathbf{x}})'\mathbf{S}^{-1}(\mathbf{x}_i - \overline{\mathbf{x}})$ where $\mathbf{x}_i$ is the $k\,x\,1$ vector of observations, $\overline{\mathbf{x}}$ is the mean vector of $\mathbf{X}$ and $\mathbf{S}$ is the $k\,x\,k$ sample covariance matrix. The robust estimates of the mean and covariance matrix are the classical mean and covariance estimates computed using a subset of the data assumed to be outlier free. The leverage methods we test are robust distances from the MVE, MCD, the *M*-estimates of covariance and the Rocke and Woodruff (1996) (R&W) hybrid estimator. We also investigate the Hadi (1992, 1994) forward search algorithm and the Sebert, Montgomery and Rollier (1998) (SM&R) clustering algorithm that can detect multiple outliers. We also consider the usual hat diagonal measure that is equivalent to the Mahalanobis distance apart from a few constants.

4.4.1 Method Description

The Hadi (1992, 1994) forward search algorithm on robust distances. This algorithm forms the initial basis of $p + 1$ clean observations from the minimum robust distances. The robust distance measure is the Mahalanobis distance computed with the median vector and covariance matrix based on the median rather than the mean. The initial basis is sequentially increased to size $h = (n + p + 1)/2$ by adding the observation

with the least robust distance calculated from the mean vector and covariance matrix of the current basis. Next, the basis is sequentially increased by the case with the lowest robust distance using the mean vector and a corrected covariance matrix from the current basis. If the lowest robust distance exceeds $\chi^2_{p,\alpha/n}$, then all observations not in the current basis are declared outliers. We use the author's *S-Plus* code.

M-estimates of covariance. Hampel (1973) first suggested *M*-estimates of covariance, but the basic paper on these estimators is attributed to Maronna (1976). Maronna addressed the problems of existence, uniqueness, asymptotic distribution and breakdown point for these estimators. We are interested in the distances in X-space for each observation defined by $z = \hat{A}(x - \hat{t})$ where $\hat{A}$ is an estimate of the $p \times p$ multivariate scatter matrix and $\hat{t}$ the multivariate location vector. Note that $(\hat{A}'\hat{A})^{-1}$ is the estimate of the covariance matrix of X. From Huber (1981), the maximum likelihood estimate of A and t is determined by solving the simultaneous equations

$$\text{ave}\{w(|z|)z\} = 0$$

$$\text{ave}(\{u|z|)zz^T - v(|z|)I_p\} = 0$$

where u, v and w are arbitrary weight functions and $\text{ave}\{\cdot\}$ is the average taken over the sample. We solve these equations using the Newton algorithm and Huber weight functions with the associated constants and correction factors as defined in the ROBETH library accessed by *S-Plus* (Marazzi, 1993). An observation is declared an outlier in X-space if the distance z exceeds the 95th percentile of simulated (1000 replicates) distances under the null hypothesis of no outliers for a specified n and p. Chapter 2 contains a detailed discussion of the algorithm.

MVE and MCD estimators. The MVE estimate of the mean is the center of the smallest ellipsoid covering at least half of the observations. The estimate of the covariance matrix is determined from these cases along with a correction factor for consistency at multivariate normal distributions. The MCD is the set of just over half of the observations with the minimum covariance matrix determinant. Cutoff values for robust distances from simulation (1000 replicates) determine whether an observation is classified as an outlier or not. There are numerous algorithms to find the MVE and MCD that provide widely varying results for the same data set; we use the recently modified genetic algorithms internal to *S-Plus 4.5* (Burns, 1992).

The R&W (1996) hybrid procedure. Rocke and Woodruff combine several results in the literature in their complex two-phase algorithm to detect multiple outliers. The output of the first phase is an estimate of multivariate location and shape. This robust estimate is determined by first partitioning the data equally into cells to minimize the impact on computational complexity. Within each cell, the observations from the MCD using Hawkins (1993) steepest descent algorithm with random restarts provide the starting point for a sequential point addition algorithm from Hadi (1992). This result is then used as a starting point for the translated bi-weight M-estimation of the mean and covariance matrices.

The second phase runs a simulation to determine the appropriate cutoff value to classify observations as outliers based on n observations in p dimensions using clean multivariate normal data in the Phase I algorithm. To increase efficiency, new location and shape matrices are formed from the set of observations below the simulated cutoff

value. The robust distance is calculated using these new location and shape matrices and compared to a $\chi^2_{p,1-\alpha}$ critical value to classify the observation as outlying or not. The authors provided their compiled $C++$ code.

The SM&R (1998) clustering algorithm. This approach uses a single-linkage clustering algorithm with Euclidean distances on the standardized predicted and standardized residual values from a least squares fit. The algorithm finds the single largest cluster, or the bulk of the data, and classifies it as the clean observations. Mojena's stopping rule forms the final clusters by splitting a cluster tree at the average of the n - 1 tree cluster heights (a measure of cluster separation) plus 1.25 times the standard deviation of the tree cluster heights.

4.4.2 Monte Carlo Simulation Leverage Study

We conduct a performance study that tests the ability of the previously described methods to identify high-leverage observations across a variety of scenarios. The key to a good leverage measure is to develop an estimate of the mean vector and covariance matrix that is not influenced by outliers. This suggests that we do not want outlying observations included in the calculations of the parameter estimates and also that we want as many clean observations included as possible. An observation is masked if it is truly an outlier but the procedure does not detect it and an observation is swamped if the procedure identifies it as an outlier when it is a clean observation. The primary measures of performance are: 1) the probability that an outlying observation is detected and 2) the probability that a known clean observation is identified as an outlier. Note that the

masking probability is the complement of the first measure of performance and the swamping probability is the second measure of performance.

The simulation scenarios place a cluster or two clusters of several observations at a specified location shifted in X-space because this geometry challenges the procedures (Rocke and Woodruff, 1996). The factors investigated in these studies are dimension, density (percentage of outlying observations), the number of standard deviations from the mean that the cloud is placed in X-space (δ_L), and the number of multiple point clouds. Additionally, we consider the number of regressors out of k that are unusual for the outlying observations as a factor because many studies only consider all k variables in their tests. Only the SM&R procedure requires response values. The response values conform to the regression surface because we do not give the SM&R procedure an unfair advantage. There are 500 replicates for each scenario and all simulations are performed in *S-Plus 4.5*.

The simulation results are reported in tables that provide the probability of detection and the probability of false alarm (in parentheses) in each cell. We also report the statistically significant effects from the analysis of variance for each procedure. The significant main effects and two factor interactions provide guidance in the table of where to look for significant differences in performance. Note that the significant effects are valid for the region of operability defined by the factor settings and a different set of effects could occur if the factor settings are changed.

4.4.2.1 Outlying Observations Unusual in All Regressor Variables

This factorial experiment tests the procedures' detection ability and resistance to swamping when the outliers are placed in a single cloud located at a distance δ_L standard deviations from the mean for all k regressor variables. The generating distribution for the clean regressor variables is N(μ_x, σ_x^2) with $\mu_x = 7.5$ and $\sigma_x = 4$. The ith observation with outlying magnitude δ_L standard deviations is placed at $x_{ij} = \overline{x}_{clean,j} + 4\delta_L + \varepsilon_{ij}^*$ for $j = 1$ to k regressor variables where $\overline{x}_{clean,j}$ is the mean of the known clean observations for the jth regressor variable. The random component ε_{ij}^*, distributed Uniform (0, 0.25), separates the observations within the outlying cloud to avoid the possibility of singular X-matrices in some procedures. An observation in a second outlying cloud (if applicable) is placed at $x_{ij} = \overline{x}_{clean,j} - 4\delta + \varepsilon_{ij}^*$. The responses for all observations are generated $y_i = \beta' x_i + \varepsilon_i$ where β is the vector of known regression coefficients selected for the simulations to be 0 for the intercept and 5 for each of the k regressor variables and ε_i is a standard normal variate.

The 2^4 factorial design in Table 4.2 contains in each cell the probability of detection and, in parentheses, the probability of false alarm. For completeness, there are four additional scenarios added to test the detection capability at higher levels of leverage (δ_L) in high-density and high-dimension scenarios because none of the procedures reliably detects the outliers at the original factor settings. The significant main effects and two-factor interactions and the average detection and false alarm probabilities that

are located in the last three rows of Table 4.2 provide summary information to assess overall performance.

There are three distinct categories of performance based on detection capability in these scenarios 1) perfect detection capability from the SM&R clustering procedure, 2) generally very good power from the MVE, MCD, R&W and the Hadi forward selection algorithm and 3) poor detection capability from the Mahalanobis distance/hat diagonal and M-estimates of covariance. The reason for the SM&R success is explained by the single-linkage clustering algorithm on the predicted and residual values. For the high-leverage observations in these runs, the OLS residuals are essentially zero and the predicted values are quite unusual with respect to the clean observations (e.g. for $k = 6$, $\delta_L = 4$, $E(y) = 225$ for the clean observations and $E(y) = 705$ for the outliers).

The combinatorial procedures (MVE, MCD, and R&W) perform well except in the high-density, high-dimension runs as indicated by the shading in Table 4.2. The four supplemental runs indicate that the R&W estimator is superior to the MVE or MCD in the high-dimension, high-density scenarios; particularly when false alarm probability is considered. We also note from Table 4.2 that all main effects, except the number of clouds, and most two-factor interactions with these three active effects, are significant for these combinatorial estimators.

The Hadi forward selection algorithm has less ability to correctly identify outliers than the combinatorial procedures. Pilot studies show power could be increased substantially (except in the high-dimension, high-density scenarios) if the cutoff value were lowered because there is a significant gap in robust distances between the clean and

Table 4.2. Design matrix with detection and false alarm probabilities (in parentheses) for high-leverage observations in multiple point clouds that have unusual values for all k regressor variables.

A n, k	B dens	C δ_L	D clds	MD	M Est Cov	MVE Chi	MVE Sim	MCD	R&W	SM&R	Hadi
40, 2	10%	3σ	1	0.000 (.000)	0.955 (.027)	0.800 (.026)	0.800 (.013)	0.746 (.011)	0.933 (.030)	1.000 (.086)	0.378 (.000)
60, 6	10%	3σ	1	0.000 (.000)	0.000 (.047)	0.900 (.038)	0.830 (.026)	0.920 (.023)	0.975 (.041)	1.000 (.090)	0.993 (.000)
40, 2	20%	3σ	1	0.000 (.011)	0.000 (.031)	0.541 (.060)	0.534 (.039)	0.647 (.041)	0.625 (.028)	1.000 (.129)	0.349 (.005)
60, 6	20%	3σ	1	0.000 (.000)	0.000 (.113)	0.000 (.462)	0.000 (.284)	0.000 (.508)	0.387 (.091)	1.000 (.079)	0.000 (.000)
40, 2	10%	4σ	1	0.953 (.001)	1.000 (.027)	0.960 (.030)	0.960 (.016)	0.990 (.010)	1.000 (.030)	1.000 (.050)	0.985 (.000)
60, 6	10%	4σ	1	0.000 (.000)	0.000 (.052)	1.000 (.035)	1.000 (.022)	1.000 (.013)	1.000 (.039)	1.000 (.027)	1.000 (.000)
40, 2	20%	4σ	1	0.000 (.021)	0.000 (.029)	0.892 (.051)	0.833 (.042)	0.927 (.031)	0.977 (.029)	1.000 (.092)	0.935 (.002)
60, 6	20%	4σ	1	0.000 (.000)	0.000 (.053)	0.040 (.332)	0.020 (.099)	0.000 (.340)	0.740 (.062)	1.000 (.063)	0.000 (.059)
40, 2	10%	3σ	2	0.140 (.032)	0.979 (.026)	0.836 (.027)	0.761 (.014)	0.773 (.009)	0.993 (.032)	1.000 (.053)	0.265 (.000)
60, 6	10%	3σ	2	0.000 (.000)	0.000 (.044)	0.985 (.038)	0.980 (.020)	0.998 (.012)	1.000 (.035)	1.000 (.066)	0.938 (.000)
40, 2	20%	3σ	2	0.000 (.037)	0.015 (.027)	0.648 (.020)	0.595 (.010)	0.670 (.007)	0.537 (.037)	0.978 (.083)	0.258 (.002)
60, 6	20%	3σ	2	0.000 (.000)	0.000 (.080)	0.000 (.123)	0.000 (.083)	0.000 (.251)	0.333 (.039)	1.000 (.042)	0.000 (.001)
40, 2	10%	4σ	2	0.899 (.032)	0.976 (.027)	0.998 (.029)	0.997 (.014)	1.000 (.009)	1.000 (.032)	1.000 (.016)	0.956 (.000)
60, 6	10%	4σ	2	0.000 (.000)	0.000 (.048)	1.000 (.035)	1.000 (.023)	1.000 (.013)	1.000 (.037)	1.000 (.009)	1.000 (.000)
40, 2	20%	4σ	2	0.000 (.024)	0.271 (.026)	1.000 (.031)	0.931 (.008)	0.988 (.006)	0.982 (.024)	1.000 (.052)	0.908 (.002)
60, 6	20%	4σ	2	0.000 (.000)	0.000 (.082)	0.040 (.122)	0.011 (.097)	0.175 (.224)	0.635 (.085)	1.000 (.031)	0.030 (.010)
60, 6	20%	5σ	1	0.000 (.000)	0.000 (.110)	0.040 (.128)	0.040 (.096)	0.015 (.341)	0.783 (.073)	1.000 (.072)	0.011 (.044)
60, 6	20%	6σ	1	0.000 (.000)	0.000 (.109)	0.230 (.113)	0.230 (.085)	0.107 (.310)	0.927 (.038)	1.000 (.042)	0.000 (.034)
60, 6	20%	5σ	2	0.000 (.000)	0.000 (.089)	0.150 (.110)	0.150 (.076)	0.305 (.195)	0.760 (.072)	1.000 (.028)	0.061 (.013)
60, 6	20%	6σ	2	0.000 (.000)	0.000 (.088)	0.370 (.095)	0.370 (.068)	0.713 (.089)	0.933 (.032)	1.000 (.027)	0.080 (.011)
Average				0.100 (.008)	0.210 (.057)	0.572 (.095)	0.552 (.057)	0.599 (.122)	0.826 (.044)	0.999 (.057)	0.457 (.009)
Significant effects Detection capability				A, B, C, AB, AC, BC	A, B, AB	A, B, C, AB, AC	A, B, C, AB, AC	A, B, C, AB, AC	A, B, C, AB, AC, BC	none	A, B, C, AB, AC
Significant effects False alarms				A, D, AD	A, B	A, B, D, AB, BD	A, B, AB	A, B, AB	A, B, AB	A, B, C, D, AB	none

outlying observations. The distances, although unusual, do not cross the threshold to declare the observations outliers. We also note the virtual nonexistence of false alarms with the Hadi method. The *M*-estimates of covariance and the hat diagonals (Mahalanobis distance) only have power in low-dimension, low-density and high-magnitude. Interestingly, these procedures still do not detect outliers in high-dimension and/or high-density if the outlying magnitude δ_L is as high as $50\sigma_x$. Although these two procedures are not useful at detection, they generally will not swamp clean observations.

4.4.2.2 Outlying Observations that are Unusual in Only One of *k* Variables

In many data sets, the high-leverage outliers may be unusual in only a single variable rather than the entire variable set as is often investigated in published data sets and in Section 4.4.2.1. This experiment is similar to Section 4.4.2.1 only the last $k-1$ regressor variables values are generated from NID(7.5, 4^2) for both the clean and outlying observations. Essentially, these are randomly scattered outliers with the cloud(s) formed only in a single regressor variable. Our experiments have shown for all methods there is a dramatic decrease in power and an increase in false alarm rate if the remaining $k-1$ variables are placed approximately at the mean of 7.5 rather than allowed to randomly vary as NID (7.5, 4^2) for the outlying observations.

Pilot studies indicate that none of the procedures have any detection capability until $\delta_L = 4\sigma_x$; therefore, the low level for leverage magnitude is increased for this study from $3\sigma_x$ to $4\sigma_x$. The design matrix and results in Table 4.3 are supplemented with two additional runs at a higher magnitude δ_L for the high-dimension, high-density runs.

The results for this section are generally consistent with those from the previous experiment in Section 4.4.2.1; however, the SM&R clustering algorithm performance has a significant decrease in detection capability and increase in false alarm rate. This can be attributed to the outliers' predicted response values not being as unusual as those of Section 4.4.2.1 because only one, rather than all, regressors is abnormally large. The SM&R method's detection capability is competitive with the others in low dimension, but has little power in high-dimension and suffers from high false alarm rates in all scenarios. The MVE, MCD and R&W procedures have lost significant detection capability (30% - 50%) from the previous study in similar scenarios at the $\delta_L = 4\sigma$ factor settings for outlying magnitude. The R&W estimator is either at or near the top in detection capability for these scenarios. In contrast to the findings in Section 4.4.2.1, we note that the MCD and MVE estimators perform reasonably well in the high-dimension, high-density runs, particularly for false alarm probabilities. The combinatorial estimators outperform the Hadi procedure. Significant gaps still exist for the Hadi procedure between the outlier and inlier robust distances and also there are no false alarms. The M-estimates of covariance and the Mahalanobis distance again are poor performers. The fact that M-estimates of covariance have more power if only a single regressor variable is outlying rather than all k is somewhat counterintuitive.

Table 4.3. Design matrix with detection and false alarm probabilities (in parentheses) for outlying multiple point clouds in only one of the k variables.

A n, k	B dens	C δ_L	D clds	MD	M Est Cov	MVE chi	MVE sim	MCD	R&W	SM&R	Hadi
40, 2	10%	4σ	1	0.193 (.021)	0.940 (.028)	0.848 (.026)	0.773 (.017)	0.793 (.010)	0.955 (.031)	0.570 (.160)	0.241 (.000)
60, 6	10%	4σ	1	0.080 (.009)	0.273 (.038)	0.589 (.030)	0.528 (.016)	0.348 (.014)	0.606 (.038)	0.182 (.150)	0.083 (.000)
40, 2	20%	4σ	1	0.000 (.010)	0.120 (.021)	0.769 (.064)	0.653 (.021)	0.685 (.014)	0.575 (.028)	0.810 (.187)	0.234 (.000)
60, 6	20%	4σ	1	0.024 (.006)	0.104 (.034)	0.123 (.028)	0.095 (.017)	0.183 (.010)	0.190 (.036)	0.091 (.172)	0.000 (.000)
40, 2	10%	5σ	1	0.325 (.036)	1.000 (.028)	0.965 (.030)	0.953 (.017)	0.970 (.001)	1.000 (.030)	0.797 (.129)	0.815 (.000)
60, 6	10%	5σ	1	0.107 (.012)	0.408 (.036)	0.877 (.033)	0.835 (.019)	0.606 (.010)	0.927 (.038)	0.263 (.142)	0.068 (.000)
40, 2	20%	5σ	1	0.000 (.030)	0.255 (.021)	0.810 (.042)	0.790 (.032)	0.904 (.028)	0.833 (.028)	0.950 (.171)	0.801 (.002)
60, 6	20%	5σ	1	0.031 (.008)	0.121 (.031)	0.386 (.021)	0.288 (.017)	0.452 (.006)	0.612 (.032)	0.135 (.166)	0.144 (.000)
40, 2	10%	4σ	2	0.012 (.035)	0.983 (.028)	0.843 (.031)	0.747 (.015)	0.701 (.012)	0.963 (.035)	0.785 (.118)	0.166 (.000)
60, 6	10%	4σ	2	0.100 (.011)	0.342 (.038)	0.570 (.032)	0.466 (.019)	0.286 (.016)	0.565 (.043)	0.227 (.144)	0.000 (.000)
40, 2	20%	4σ	2	0.000 (.035)	0.375 (.022)	0.551 (.033)	0.521 (.013)	0.574 (.010)	0.382 (.035)	0.730 (.177)	0.163 (.000)
60, 6	20%	4σ	2	0.036 (.009)	0.141 (.033)	0.201 (.025)	0.141 (.014)	0.182 (.013)	0.195 (.034)	0.090 (.172)	0.000 (.000)
40, 2	10%	5σ	2	0.380 (.032)	1.000 (.026)	0.962 (.021)	0.933 (.016)	0.938 (.009)	1.000 (.032)	0.955 (.080)	0.781 (.000)
60, 6	10%	5σ	2	0.138 (.015)	0.532 (.037)	0.937 (.032)	0.831 (.020)	0.567 (.011)	0.940 (.040)	0.360 (.140)	0.108 (.000)
40, 2	20%	5σ	2	0.000 (.029)	0.739 (.022)	1.000 (.035)	0.878 (.009)	0.930 (.006)	0.877 (.029)	0.885 (.127)	0.764 (.002)
60, 6	20%	5σ	2	0.043 (.011)	0.157 (.030)	0.579 (.021)	0.506 (.014)	0.445 (.006)	0.626 (.027)	0.233 (.160)	0.085 (.000)
60, 6	20%	6σ	1	0.031 (.008)	0.130 (.031)	0.727 (.023)	0.703 (.013)	0.838 (.006)	0.939 (.024)	0.185 (.182)	0.643 (.000)
60, 6	20%	6σ	2	0.046 (.011)	0.168 (.029)	0.823 (.023)	0.804 (.011)	0.834 (.020)	0.829 (.008)	0.983 (.027)	0.591 (.000)
Average				0.092 (.019)	0.468 (.030)	0.688 (.032)	0.621 (.017)	0.598 (.011)	0.703 (.034)	0.504 (.150)	0.278 (.000)
Significant effects detection capability				B	A, B, AB	A, B, C, AB	A, B, C, AB	A, B, C	A, B, C, AC, BC	A, C, AB	A, C, AC
Significant effects false alarm				A	A, B	AB	D, AD, BD	none	A, B	B, C, D	none

4.4.2.3 High-density, High-Magnitude Outliers

The results from the previous two studies indicate that the procedures have difficulty correctly identifying the outliers in the high-dimension, high-density scenarios. This study changes the levels for the total outlier density factor from 10% for the low

level and 20% for the high level to 15% and 30% respectively. The levels for the

distance the cloud is shifted, δ_L, are also changed from 3σ and 4σ to 5σ and 10σ,

respectively. Because the number of clouds is generally not a significant factor

contributing to the performance of these procedures, it is set to a constant value of one for

all scenarios. The fourth factor is now the number of regressor variables out of k that

have outlying values for the planted outliers. The low setting is one and the high setting

is all k variables (2 or 6). The Mahalanobis distance has no power in virtually all

scenarios; therefore, its performance is not included with the results in any further

studies.

The most interesting result from this study is the breakdown of the procedures at

30% density shown in the shaded high-dimension scenarios of Table 4.4. The false alarm

rates are abnormally large when the values are extreme in all k variables for these runs

and also the shaded run in low dimension. The SM&R clustering procedure performance

is consistent with previous findings; 100% detection capability if outlying in all regressor

variables and a significant loss of power if outlying only in a single variable. Similarly,

the M-estimates of covariance have detection power limited exclusively to low dimension

and low-density scenarios independent of the magnitude of the outlying distance. The

R&W hybrid estimator is typically more powerful than the MCD or MVE. Surprisingly,

only a single factor (contamination percentage) is significant for detection capability for

R&W and none are significant for the MCD in this operating region.

Table 4.4. Design matrix with detection and false alarm probabilities (in parentheses) for high-magnitude, high-density, high-leverage scenarios. Outliers are unusual in one or all regressors.

A n, k	B dens	C δ_L	D vars	M Est Cov	MVE chi	MVE Sim	MCD	R&W	SM&R	Hadi
40, 2	15%	5σ	1	0.920 (.029)	0.933 (.033)	0.915 (.018)	0.940 (.009)	0.965 (.034)	0.745 (.171)	0.835 (.000)
60, 6	15%	5σ	1	0.200 (.033)	0.751 (.030)	0.694 (.017)	0.608 (.009)	0.860 (.032)	0.191 (.152)	0.101 (.000)
40, 2	30%	5σ	1	0.058 (.017)	0.625 (.016)	0.602 (.005)	0.928 (.002)	0.633 (.018)	0.449 (.273)	0.793 (.000)
60, 6	30%	5σ	1	0.071 (.039)	0.052 (.030)	0.030 (.016)	0.358 (.011)	0.234 (.038)	0.062 (.227)	0.141 (.000)
40, 2	15%	10σ	1	1.000 (.031)	1.000 (.031)	1.000 (.016)	1.000 (.009)	1.000 (.034)	1.000 (.077)	1.000 (.000)
60, 6	15%	10σ	1	0.293 (.031)	1.000 (.031)	1.000 (.019)	1.000 (.008)	1.000 (.032)	0.856 (.138)	1.000 (.000)
40, 2	30%	10σ	1	0.680 (.017)	1.000 (.015)	1.000 (.003)	1.000 (.002)	0.977 (.020)	1.000 (.183)	0.944 (.002)
60, 6	30%	10σ	1	0.066 (.035)	0.153 (.078)	0.128 (.015)	1.000 (.003)	0.963 (.022)	0.534 (.281)	0.939 (.000)
40, 2	15%	5σ	2	1.000 (.036)	1.000 (.029)	1.000 (.015)	1.000 (.009)	1.000 (.034)	1.000 (.054)	1.000 (.000)
60, 6	15%	5σ	6	0.000 (.070)	0.840 (.038)	0.840 (.022)	0.860 (.052)	0.970 (.037)	1.000 (.035)	1.000 (.000)
40, 2	30%	5σ	2	0.000 (.064)	0.270 (.276)	0.270 (.244)	0.300 (.302)	0.807 (.092)	1.000 (.128)	1.000 (.000)
60, 6	30%	5σ	6	0.000 (.143)	0.000 (.339)	0.000 (.280)	0.000 (.621)	0.010 (.260)	1.000 (.109)	0.000 (.474)
40, 2	15%	10σ	2	1.000 (.036)	1.000 (.034)	1.000 (.017)	1.000 (.009)	1.000 (.034)	1.000 (.034)	1.000 (.000)
60, 6	15%	10σ	6	0.000 (.068)	1.000 (.027)	1.000 (.016)	1.000 (.009)	1.000 (.032)	1.000 (.028)	1.000 (.000)
40, 2	30%	10σ	2	0.000 (.063)	0.930 (.053)	0.930 (.040)	0.980 (.015)	0.990 (.024)	1.000 (.112)	1.000 (.000)
60, 6	30%	10σ	6	0.000 (.142)	0.000 (.337)	0.000 (.281)	0.000 (.622)	0.400 (.189)	1.000 (.089)	0.000 (.453)
Average				0.331 (.053)	0.660 (.087)	0.651 (.064)	0.748 (.106)	0.801 (.058)	0.802 (.131)	0.735 (.058)
Significant effects detection capability				A, B, AB	A, B, C, AB	A, B, C, AB	none	B	C, D, CD	A
Significant effects false alarm				A, B, D, AB, AD, BD	B, D, AB, BD	B, D, BD	B, D, BD	None	C, D	none

4.4.2.4 Outlying Observations with Unusual Levels in 3 of 6 Variables

This section investigates the difference in detection capability and false alarm rates for the procedures when the outliers have an intermediate factor setting of outlying in 3 of 6 regressor variables. The motivation is the discrepancy in performance when the outliers are unusual in all 6 variables versus outlying in only 1 of 6. There are only three

factors to consider in this study because the dimension is set at $k = 6$ and the number of

outlying variables is set at 3. The results are shown for the full factorial in three factors

in Table 4.5.

From Table 4.5, the overall detection probabilities are similar to those seen in the

runs with all 6 variables outlying. However, the false alarm averages in the high-density

scenarios are much lower compared to the rates when outlying in all 6 variables. The

exception to this is the SM&R false alarm rates near 20% for many of the high-density

scenarios. Again, the R&W hybrid algorithm slightly outperforms the MCD and MVE in

most cases. The MVE is vulnerable in the high outlier density scenarios for the 4 and 5σ

Table 4.5. Design matrix with detection and false alarm probabilities
(in parentheses) for clouds that are remote in 3 of the 6 regressor variables.

A Dens	B δ_L	C clds	M Est Cov	MVE chi	MVE Sim	MCD	R&W	SM&R	Hadi
10 %	4σ	1	0.330 (.041)	0.990 (.033)	0.990 (.018)	0.954 (.012)	1.000 (.018)	0.998 (.134)	0.958 (.000)
20 %	4σ	1	0.040 (.052)	0.327 (.050)	0.245 (.034)	0.813 (.028)	0.819 (.040)	0.968 (.253)	0.920 (.000)
10%	5σ	1	0.517 (.042)	0.980 (.033)	0.967 (.023)	1.000 (.013)	1.000 (.037)	1.000 (.055)	0.991 (.004)
20 %	5σ	1	0.042 (.049)	0.597 (.035)	0.591 (.022)	0.965 (.011)	0.946 (.031)	1.000 (.220)	0.987 (.000)
10%	4σ	2	0.510 (.040)	1.000 (.033)	0.919 (.023)	0.956 (.013)	1.000 (.037)	1.000 (.048)	0.903 (.000)
20 %	4σ	2	0.053 (.043)	0.583 (.036)	0.550 (.023)	0.935 (.006)	0.958 (.029)	0.964 (.194)	0.853 (.000)
10%	5σ	2	0.785 (.041)	0.990 (.034)	1.000 (.025)	1.000 (.013)	1.000 (.037)	1.000 (.027)	1.000 (.000)
20%	5σ	2	0.054 (.042)	0.837 (.033)	0.813 (.019)	0.999 (.007)	0.990 (.029)	1.000 (.056)	1.000 (.000)
20%	6σ	1	0.043 (.049)	0.793 (.032)	0.792 (.018)	0.810 (.027)	0.867 (.041)	1.000 (.076)	0.972 (.000)
30%	10σ	1	0.286 (.043)	0.063 (.103)	0.059 (.071)	0.870 (.060)	0.891 (.039)	0.995 (.095)	0.237 (.006)
Average			0.266 (.044)	0.716 (.042)	0.693 (.028)	0.930 (.019)	0.947 (.034)	0.993 (.116)	0.882 (.001)
Significant effects Detection capability			A	A	A	B	none	B, A, AB	B
Significant effects false alarm			A, C, AC	none	none	None	none	A	None

cases, but is competitive with the other combinatorial procedures by 6σ and beyond. The last scenario in Table 4.5 again shows the vulnerability of the MVE and the breakdown of the otherwise excellent performing Hadi forward search.

4.4.2.5 Outlying Observations Without Unusual Response Values

This study evaluates the performance when the response value for the outlying observations is not a Y-space outlier. The purpose of this study is twofold. First, interesting results can occur when the signs of regressor variables are changed, as we do here, and second, to investigate the effect on the SM&R algorithm that has performed well with unusual predicted response values. Recall that the regressor variables for the clean observations are generated from a $N(7.5, 4^2)$ distribution. In the studies to this point, an observation in an outlying cloud at, for example, $\delta_L = 4\sigma_x$ in two regressor variables would be placed at $x1 = x2 = 7.5 + 4(4) + \varepsilon_{ij}^* \cong 23.5$ where ε_{ij}^* is distributed Uniform $(0, 0.25)$. The expected response value would be approximately $5*23.5 + 5*23.5 = 235$. The expected response value for clean observation is significantly lower, $5* 7.5 + 5*7.5 = 75$. For the scenarios in this experiment, the outlying cloud is placed approximately $4\sigma_x$ above the mean or 23.5 for x1 and $4\sigma_x$ below the mean or -8.5 for x2. The expected response for the outliers, $5 * 23.5 + 5 * (-8.5) = 75$, is now the same as that for the clean observations. The scenarios selected for the study in Table 4.6 are random; however, the results are consistent independent of the factor settings.

The results indicate that SM&R has no power to detect outliers in X-space if the response variable is not unusual and the least squares residuals are driven essentially to

zero by the high-leverage points. The 20% false alarm rate for this method is consistent with the published value for the null behavior. The other methods perform slightly below their counterpart runs when all variables have the same sign.

Table 4.6. Design matrix with detection and false alarm probabilities (in parentheses) for high-leverage points without unusual response values.

n, k	dens	δ_L	clds	Vars	M Est Cov	MVE Chi	MVE Sim	MCD	R&W	SM&R	Hadi
40, 2	20%	4σ	1	2	0.000 (.032)	0.977 (.082)	0.883 (.026)	0.958 (.021)	0.980 (.025)	0.005 (.199)	0.959 (.003)
40, 2	10%	$3\sigma, 4\sigma$	2	2	0.965 (.029)	0.975 (.031)	0.909 (.012)	0.871 (.009)	0.993 (.028)	0.005 (.201)	0.405 (.000)
60, 6	20%	3σ	2	6	0.000 (.069)	0.893 (.064)	0.864 (.027)	0.932 (.210)	0.926 (.039)	0.000 (.200)	0.087 (.014)
60, 6	20%	$3\sigma, 6\sigma*$	2	3	0.000 (.086)	0.775 (.059)	0.740 (.029)	0.905 (.020)	0.915 (.034)	0.000 (.199)	0.029 (.005)
Average					0.241 (.054)	0.905 (.059)	0.849 (.024)	0.917 (.065)	0.954 (.032)	0.003 (.200)	0.370 (.006)

*This scenario has the outliers at x1=3σ, x2 = 3σ and x3= -6σ for the first cloud and x1=-3σ, x2 = -3σ and x3= 6σ for the second cloud.

4.4.2.6 Multiple Point Clouds that are in Close Proximity

These miscellaneous scenarios test the ability to identify outlying multiple point clouds positioned next to each other in X-space. In Sections 4.4.2.1 through 4.4.2.4, if there are two multiple point clouds, one is placed at $+ \delta_L\sigma_x$ and the other at $- \delta_L\sigma_x$. The location of the point clouds for this study is specified in the outlying magnitude column of Table 4.7. These scenarios have been cited as challenging in the literature because the outlying clouds can mask the other clouds from detection.

The overall results are mixed. The SM&R detection capability is consistent with earlier results: excellent detection capability if outlying in 3 of 6 or all 6 variables, poor detection capability if outlying in a single variable, and high false alarm rates. The shaded scenarios highlight that R&W outperforms or is competitive with the MVE and

MCD. Note, in particular, the uncharacteristic high false alarm rate for the MCD procedure in these three shaded scenarios.

Table 4.7. Design matrix with detection and false alarm probabilities (in parentheses) for multiple point clouds located in close proximity.

n, k	dens	δ_L	clds	vars	M-est cov	MVE Chi	MVE sim	MCD	R&W	SM&R	Hadi
40, 2	10%	$3\sigma, 4\sigma$	2	2	0.976 (.028)	0.931 (.046)	0.883 (.016)	0.841 (.012)	0.995 (.035)	1.000 (.088)	0.608 (.000)
40, 2	20%	$3\sigma, 4\sigma$	2	2	0.000 (030)	0.723 (.058)	0.670 (.033)	0.749 (.031)	0.623 (.032)	0.970 (.133)	0.571 (.000)
40, 2	10%	$3\sigma, 4\sigma$	2	1	0.685 (.026)	0.644 (.029)	0.526 (.019)	0.434 (.013)	0.786 (.032)	0.439 (.154)	0.021 (.000)
40, 2	20%	$3\sigma, 4\sigma$	2	1	0.131 (.021)	0.384 (.021)	0.283 (.013)	0.311 (.006)	0.171 (.022)	0.222 (.219)	0.019 (.000)
60, 6	10%	$3\sigma, 4\sigma$	2	6	0.005 (.053)	0.995 (.030)	0.995 (.017)	0.987 (.014)	0.960 (.040)	1.000 (.043)	0.997 (.000)
60, 6	20%	$3\sigma, 4\sigma$	2	6	0.000 (.078)	0.000 (.131)	0.000 (.094)	0.000 (.328)	0.650 (.071)	1.000 (.093)	0.090 (.051)
60, 6	10%	$3\sigma, 4\sigma$	2	3	0.382 (.042)	0.965 (.031)	0.918 (.020)	0.783 (.014)	0.956 (.037)	0.896 (.144)	0.428 (.000)
60, 6	20%	$3\sigma, 4\sigma$	2	3	0.048 (.053)	0.276 (.050)	0.193 (.030)	0.563 (.035)	0.643 (.044)	0.755 (.244)	0.398 (.000)
60, 6	10%	$4\sigma, 5\sigma$	2	1	0.385 (.038)	0.768 (.033)	0.622 (.017)	0.414 (.014)	0.701 (.039)	0.182 (.153)	0.014 (.000)
60, 6	20%	$4\sigma, 5\sigma$	2	1	0.123 (.033)	0.248 (.029)	0.185 (.017)	0.271 (.011)	0.202 (.040)	0.095 (.173)	0.000 (.000)
60, 6	20%	$3\sigma, 4\sigma, 5\sigma, 6\sigma$	4	6	0.000 0.071	0.240 (.122)	0.215 (.102)	0.020 (.308)	0.753 (.068)	1.000 (.196)	0.217 (.001)
60, 6	20%	$3\sigma, 4\sigma, 5\sigma, 6\sigma$	4	3	0.056 (.050)	0.636 (.043)	0.628 (.023)	0.823 (.012)	0.807 (.033)	0.893 (.239)	0.457 (.000)
60, 6	20%	$3\sigma, 4\sigma, 5\sigma, 6\sigma$	4	1	0.167 (.034)	0.326 (.021)	0.368 (.021)	0.484 (.020)	0.283 (.040)	0.178 (.164)	0.000 (.000)
60, 6	20%	$-5\sigma, -4\sigma, 4\sigma, 5\sigma$	4	6	0.000 0.073	0.120 (.119)	0.070 (.098)	0.240 (.215)	0.680 (.076)	1.000 (.143)	0.405 (.004)
60, 6	20%	$-5\sigma, -4\sigma, 4\sigma, 5\sigma$	4	3	0.058 (.044)	0.797 (.053)	0.769 (.037)	0.995 (.019)	0.980 (.041)	0.978 (.186)	0.965 (.000)
60, 6	20%	$-5\sigma, -4\sigma, 4\sigma, 5\sigma$	4	1	0.163 (.032)	0.368 (.023)	0.277 (.013)	0.332 (.005)	0.254 (.031)	0.235 (.163)	0.013 (.000)
60, 6	20%	$-6\sigma, -5\sigma, 5\sigma, 6\sigma$	4	1	0.184 (.031)	0.660 (.109)	0.653 (.011)	0.682 (.005)	0.788 (.027)	0.286 (.162)	0.193 (.000)
Average					0.198 (.043)	0.534 (.056)	0.486 (.034)	0.525 (.062)	0.661 (.042)	0.655 (.159)	0.317 (.003)

4.4.3 Summary of Performance for Measures of Leverage

This section summarizes the key findings from the comparative evaluation for each technique. There was no need to test the Mahalanobis distance (hat diagonal) after the first few experiments because it has no power to detect multiple outliers in X-space except under some specific conditions.

M-estimates of covariance. The usefulness of the *M*-estimates of covariance distances to detect multiple outliers in X-space is limited to low-dimension, low-density scenarios. In high-dimension, the outlying leverage distance, δ_L, can be increased without bound; yet, the *M*-estimates of covariance distances for the planted outliers are still not unusual compared to the inlier distances. False alarm probabilities are always below the nominal 5% rate unless in high-dimension, high-density. Overall, this method is only slightly preferable to the Mahalanobis distance and it is inferior to the other methods to identify outliers in X-space.

Hadi. The Hadi forward selection algorithm is shown to have decent detection capability with very low false alarm probabilities. Although often outperformed by other procedures, significant improvement in detection capability is possible by lowering the cutoff values. The procedure does not perform well in high-dimension, high-density. Further experimentation shows little improvement in high-dimension, high-density scenarios if we modify the cutoff values.

SM&R. The clustering algorithm of the least squares standardized predicted and residual values is most effective when the predicted values are unusual with respect to the response values for the clean observations. The procedure has higher detection

probabilities if the number of unusual regressors is large. The usefulness of this procedure as an input to the measure of leverage is limited by the large false alarm rates in many scenarios and the general requirement for unusual predicted response values.

MCD. The robust distances from the MCD estimates are a dependable measure of leverage and useful to detect multiple outliers. The simulated cutoff values for robust distances from the genetic algorithm in *S-Plus* allow reasonable detection capability and limit the impact from false alarms. The MCD and MVE have similar performance. The MCD occasionally will significantly outperform the MVE, especially if $k = 6$ and the number of outlying regressors is 3 or more. There are several high-dimension, high-density scenarios where both the MVE and MCD fail to detect the outliers highlighted throughout Section 4.4.2. In these instances, the MCD has a much greater false alarm probability than the MVE or any other procedure. These are the only instances when the MCD procedure exceeds the nominal 5% false alarm rate.

MVE. Robust distances based on the MVE estimate are shown to reliably detect high-leverage observations in most scenarios. Overall, the MVE from the genetic algorithm is competitive with the other combinatorial estimators (MCD and R&W), but can have significantly lower detection capability in high-dimension, high-density scenarios. The use of chi-square critical values for the robust distances has a consistently high false alarm rate. Therefore, a moderate decrease in detection capability from the simulated critical values is a worthwhile tradeoff to control false alarms.

Rocke and Woodruff. This hybrid procedure consistently performed as well as or better than the MVE and MCD in both our studies and those of the authors. The one

exception is that the MCD has slightly better detection probabilities in low-dimension, high-density. R&W has superior performance to the MVE and MCD in the challenging high-dimension, high-density scenarios. In fact, there are only two scenarios in Section 4.4.2.3 that the procedure fails to detect the outliers and significantly swamps clean observations. The results suggest that this is the preferred procedure to identify outliers in X-space. Of course, if the outlying observations are known to have unusual response values, the SM&R algorithm is recommended.

The R&W procedure is the most versatile and often the best performer in our studies and we suggest its use as a measure of leverage. Rocke and Woodruff (1997) cite instances, most notably in high-dimension, when the MVE and MCD procedures alone are likely to fail where theirs will not. They also note that their procedure is much more computationally efficient for high-dimension and large sample size compared to the MCD and especially the MVE. Our experience has shown reasonable computation times for $p \leq 10$, $n \leq 100$ on a modest PC (PII-350, 96 MB RAM) when implemented through a dynamically linked library (DLL) in *S-Plus 4.5*. However, the computation times for the MVE, MCD or *M*-estimates of covariance are significantly less than those of R&W.

4.5 Compound Estimators with R&W Robust Distances as the π-weight Component

The simulation results in the previous section indicate that the *M*-estimates of covariance used in the S&M compound estimator and the robust distances using the MVE estimator with χ^2 cutoff values used in C&H are not the strongest performing techniques. The best performing alternative is the R&W robust distances. The R&W

robust distances are especially powerful in high-dimension, high density scenarios. We now evaluate the utility of using the R&W robust distances as the component in the π-weights for the S&M and C&H estimators.

Example 4.1 Revisited. Although the exact factor settings in Example 4.1 are not run in Section 4.4, the results from Table 4.6 with $n = 60$, $k = 6$, $\delta_L = 5\sigma_x$ and outlying in 3 (rather than 2 as in our example) of the 6 variables provide an accurate indication of performance. The M-estimates of covariance have virtually no power to detect the leverage points and the MVE robust distances are only about 60% effective. However, the R&W robust distances detect the outliers approximately 95% of the time. The modified S&M estimator replaces the M-estimates of covariance distances with the R&W distances. The modified C&H estimator replaces the MVE robust distances with the R&W robust distances. The C&H π-weights also use a chi-square cutoff value; we replace it with the R&W cutoff value supplied by the algorithm.

Unfortunately, the parameter estimates and residuals for both modified estimators have changed little from their original values with this leverage modification. On a positive note, many of the final weights for the outliers have changed to a value between 0.0 and 0.5 as opposed to 0.99 in the original versions. Also, the π-weights are now unusual for the 12 outlying observations.

Further experimentation with modified S&M and C&H techniques indicates there is no real advantage to the improved π-weights in virtually all high-leverage scenarios. For both the original and modified versions of these two compound estimators, we observe that in the high-leverage, high-dimension scenarios the final parameter estimates

do not change much from the initial (S or LTS) values. Therefore, if the initial estimate is improved, then the compound estimators may have a better chance to accommodate these high-leverage outliers.

4.6 A Proposal for a New Initial Estimator

The high-breakdown initial estimators typically used in a compound or GM technique do not have the bounded-influence property and thus are known to have difficulty (i.e. poor parameter estimates) when there are high-leverage regression outliers. We consider a high-breakdown "rejection plus" alternative as an initial estimator that first locates and then eliminates the three classes of outliers from the data set followed by parameter estimation from a least squares fit on the reduced data set. It is important to clarify that we use the full data set for sequential stages of the compound estimation scheme.

For the proposed initial estimator, high-leverage outliers are first eliminated, without regard to how well they fit the regression surface, if the R&W robust distance exceeds the algorithm's internally calculated cutoff value. The remaining observations should all be on the interior of X-space. From Chapter 3 and Simpson and Montgomery (1998b), an excellent high-breakdown, high-efficiency estimator for low-leverage outliers is the MM estimator (Yohai, 1987 and Yohai et al., 1991). The MM estimator has three stages. The initial estimate is a high-breakdown estimate using an S-estimate. The second stage computes an M-estimate of the errors' scale from the initial S-estimate residuals. The last step is an M-estimate of the regression parameters with a

redescending ψ-function. If the absolute value of the standardized residuals from an

MM estimate on the remaining interior X-space data exceeds a simulated cutoff value,

then these observations are also removed. The simulated cutoff value is 1.91 for both

n=60, k = 6 and n = 40, k = 2 based on the 95th percentile of the residuals (absolute

values) from 1000 replications of uncontaminated data. In practice, it is probably

reasonable to use a rule of thumb of 2.0 to avoid the added complexity. The parameter

estimates for the proposed initial estimator come from a least squares fit on the remaining

observations after the two high-breakdown filters remove the leverage and residual

outliers. Therefore, the steps of the proposed initial estimator are 1) remove high-

leverage observations if the R&W robust distance exceeds the algorithm's internally

calculated cutoff value, 2) from the remaining observations, remove the residual outliers

if the *MM* residual exceeds the simulated cutoff value and 3) obtain parameter estimates

with an OLS fit on the remaining data.

This type of estimator is termed a "rejection-plus" estimator because it eliminates

outlying observations from the data and uses an optimal estimator (OLS) on the supposed

remaining clean data. The "rejection plus" regression estimator logic has been suggested

in the literature with different robust regression estimators and multiple outlier detection

algorithms (e.g. Rousseeuw, 1984, Simonoff, 1991, Hadi and Simonoff, 1993, and

Wilcox, 1997). He and Portnoy (1992) point out that the estimate of the standard error

may not converge to the correct value as n gets large for these procedures. The "rejection

plus" scheme has not been proposed as an initial estimator for multi-staged techniques.

There is less of a concern about convergence because the full data set is used in the

remaining stages of the *GM* or compound estimate. This initial estimator is appealing because the simulation results from Chapter 3 and Section 4.4 indicate both R&W and *MM* are powerful across a comprehensive set of scenarios, yet do not have a tendency to false alarm. The false alarms in this scheme impact the efficiency of the initial estimator because each false alarm results in the removal of a clean observation from the final subset used in the OLS estimate of the parameters. Another advantage for this type of initial estimator is that it is likely to be more efficient than the other high-breakdown estimators that use, in some cases, only half of the observations to estimate parameters.

4.6.1 Initial Estimator Performance Studies

The proposed initial estimator is tested against other high-breakdown initial estimators in the literature. We consider the LMS, LTS (set to achieve 30% breakdown), and *S*-estimators in addition to three variants of the proposed estimator. The first variant of the proposed initial estimator, P1, is the one previously described—an R&W filter of high-leverage points followed by an *MM* filter of residual outliers and then an OLS fit to the remaining observations. The second proposal is the P2 estimator with parameter estimates from the *MM* fit after the R&W filter rather than the OLS parameter estimates used in P1. The P3 estimator is the same as P2 except that an *S*-estimator is used in place of the *MM* estimator.

The performance study evaluates the effect on the initial estimators from multiple high-leverage and residual outliers. The factors are dimension (number of regressors), outlier density, leverage (outlying magnitude in X-space, δ_L), residual magnitude

(outlying distance off the regression surface, δ_R), and the number of regressor variables out of k that are unusual in X-space. The response, efficiency ratio, is $MSE_{clean}/MSE_{estimator}$ where MSE_{clean} is the MSE for the known clean observations from an OLS fit with only the known clean observations. $MSE_{estimator}$ is the MSE for the known clean observations from the fit using the selected estimator on the entire data set. Table 4.8a shows the 2_V^{5-1} design matrix and average efficiency ratios from 50 replicates in *S-Plus 4.5*. Note that additional replication does not significantly change the values in Table 4.8a and does not change the key findings.

Table 4.8a. Design matrix and efficiency ratios for common initial estimators.

A n, k	B density	C δ_L	D δ_R	E out	OLS	LMS	LTS	S	P1	P2	P3
40, 2	10 %	5	5	2	0.689	0.732	0.785	0.803	0.966	0.984	0.861
60, 6	10 %	5	5	1	0.631	0.606	0.719	0.734	0.922	0.960	0.698
40, 2	20 %	5	5	1	0.445	0.624	0.633	0.618	0.956	0.976	0.827
60, 6	20 %	5	5	2	0.432	0.563	0.594	0.624	0.928	0.967	0.688
40, 2	10 %	10	5	1	0.729	0.720	0.765	0.762	0.958	0.981	0.809
60, 6	10 %	10	5	2	0.706	0.626	0.709	0.710	0.949	0.977	0.716
40, 2	20 %	10	5	2	0.503	0.739	0.790	0.795	0.960	0.975	0.796
60, 6	20 %	10	5	1	0.390	0.585	0.668	0.672	0.939	0.967	0.665
40, 2	10 %	5	10	1	0.403	0.787	0.870	0.872	0.959	0.981	0.809
60, 6	10 %	5 ·	10	2	0.339	0.687	0.782	0.798	0.949	0.977	0.715
40, 2	20 %	5	10	2	0.204	0.737	0.870	0.825	0.960	0.975	0.796
60, 6	20 %	5	10	1	0.150	0.646	0.812	0.820	0.941	0.969	0.666
40, 2	10 %	10	10	2	0.514	0.749	0.792	0.806	0.966	0.984	0.861
60, 6	10 %	10	10	1	0.360	0.631	0.732	0.745	0.922	0.960	0.698
40, 2	20 %	10	10	1	0.198	0.655	0.648	0.643	0.963	0.982	0.833
60, 6	20 %	10	10	2	0.203	0.560	0.604	0.585	0.928	0.967	0.688
Average efficiency					0.431	0.665	0.736	0.738	0.948	0.974	0.758
Significant Effects					A, B, D	A, B, CD	A, B, C, D, AE, CD	A, B, C, D, AE, CD	A, E, CD, BE	A, B, E, AE, BE	A, B, E, BE, CD

As expected, the ability of the common initial estimators (LMS, LTS and *S*) to fit the clean data is significantly impacted in the high-leverage scenarios tested in Table 4.8. The LTS and *S* estimators consistently outperform LMS. The proposed initial estimators

P1 and P2 have much better efficiency ratios than the existing procedures and P3. A surprising result is that the S-estimator and P3 have similar results. Removing the high-leverage observations apparently has little effect on the S-estimator's performance in these selected scenarios. From Table 4.8a, P1 and P2 are the preferred alternatives.

In all of the scenarios in Table 4.8a, the R&W estimator detected the planted outliers because they were extreme in X-space. We now consider the performance of the initial estimators when the leverage distance δ_L is not as great. The R&W estimator does not necessarily detect and remove all of the planted observations unusual in X-space. Table 4.8b includes not only the 2_V^{5-1} design and resulting efficiency ratios, but also (in the last column) the proportion of outlying observations that R&W removes. Although there is less of a discrepancy in efficiency ratios between the first two proposed initial estimators and the alternatives, P1 and P2 again have the best results. P1 slightly outperforms P2 in these scenarios.

Table 4.8b. Design matrix and efficiency ratios for common initial estimators when R&W does not necessarily detect the leverage points. The last column is the proportion of planted outliers removed by the R&W filter.

A n, k	B dens	C δ_L	D δ_R	E out	OLS	LMS	LTS	S	P1	P2	P3	% RW
40, 2	10 %	1.5	5	2	0.640	0.825	0.875	0.904	0.968	0.976	0.877	0.50
60, 6	10 %	1.5	5	1	0.606	0.655	0.787	0.782	0.921	0.949	0.734	0.14
40, 2	20 %	1.5	5	1	0.413	0.840	0.919	0.921	0.869	0.705	0.911	0.07
60, 6	20 %	1.5	5	2	0.366	0.700	0.841	0.864	0.748	0.682	0.804	0.17
40, 2	10 %	3	5	1	0.681	0.763	0.851	0.854	0.956	0.975	0.807	0.92
60, 6	10 %	3	5	2	0.599	0.647	0.752	0.763	0.937	0.966	0.715	0.99
40, 2	20 %	3	5	2	0.436	0.661	0.685	0.683	0.956	0.969	0.795	0.99
60, 6	20 %	3	5	1	0.376	0.564	0.740	0.715	0.631	0.643	0.728	0.17
40, 2	10 %	1.5	10	1	0.376	0.791	0.871	0.874	0.961	0.982	0.843	0.16
60, 6	10 %	1.5	10	2	0.294	0.679	0.781	0.794	0.940	0.974	0.751	0.24
40, 2	20 %	1.5	10	2	0.159	0.827	0.922	0.913	0.966	0.987	0.859	0.32
60, 6	20 %	1.5	10	1	0.141	0.656	0.829	0.838	0.938	0.972	0.816	0.10
40, 2	10 %	3	10	2	0.347	0.827	0.884	0.904	0.966	0.985	0.858	1.00
60, 6	10 %	3	10	1	0.310	0.655	0.785	0.783	0.931	0.966	0.732	0.36
40, 2	20 %	3	10	1	0.247	0.787	0.885	0.884	0.957	0.972	0.825	0.69
60, 6	20 %	3	10	2	0.149	0.713	0.857	0.866	0.932	0.972	0.716	0.80
Average efficiency					0.384	0.724	0.829	0.883	0.911	0.917	0.798	
Significant Effects					A, B, D	A, C CD	A, C, CD	A, C, CD	A, B, D, BD	B, D, BD	A, C	

4.7 Proposal of New Compound Estimators

The results of Sections 4.4 and 4.6 suggest components of the S&M and C&H compound estimators could be changed to increase the envelope of effective performance in high-leverage, high-dimension scenarios. Section 4.4 clearly indicates superior performance of the R&W robust distances over the *M*-estimates of covariance distances and moderately better performance over the MVE robust distances. Section 4.6 indicates that an improved high-breakdown initial estimator that can accommodate high-leverage outliers is possible by using P1 or P2 rather than the existing LTS and *S*-estimators. Therefore, the components of the proposed compound estimator CEP1 are a P1 initial estimate, R&W robust distances as the measure of leverage, π-weights as the ratio of R&W robust distance to the median robust distance, an LMS estimate of scale using the

residuals from the P1 initial estimate, a Tukey bi-weight ψ function with tuning constant 4.685 for 95% efficiency, and a one-step convergence from IRLS. CEP1 is similar to the S&M estimator except that the measure of leverage is R&W versus the M-estimates of covariance distances and the initial estimate is P1 versus an S-estimate. Another difference is the estimate of scale. S&M uses the S-estimate of scale in part because it is readily available from the initial S-estimate. Rather than add even further computational complexity to our procedure, we use the LMS estimate of scale defined as $1.4826 * (1 + 5/(n - p)) * \text{median } |e_{P1}|$ where e_{P1} is the vector of residuals from the initial fit with the P1 estimator. The second proposed compound estimator, CEP2, is the same as CEP1 except that the initial estimate is from P2.

We also consider the proposed compound estimators CEP3 and CEP4 that are the S&M estimator with R&W leverage measures modified by using 3 or 4 iterations, respectively, of IRLS to solve the normal equations. The motivation for these estimators comes from the improvement in final weights in the example problem. Also, He and Portnoy (1992) suggest that in practice a single step of a GM iteration scheme is often insufficient. Pilot studies showed that the parameter estimates do not change significantly after 2 iterations and that at least 3 are required to effectively accommodate the high-leverage outliers across a variety of test scenarios. Simpson and Chang (1997) demonstrate that several iterations still maintain the same first order large sample properties as the single iteration version of the compound estimator.

Example 4.1 provides an initial performance indication of the four proposed compound estimators. All four estimators effectively identify and downweight the 12

planted outliers in this example. Their parameter estimates are similar to each other.

These estimates are much closer to both the generating β vector and the estimates from

OLS fit on the known 48 clean observations. Additionally, the initial estimate for CEP1

and CEP2 is efficient because the correct 12 outlying observations are rejected and 44 out

of the 48 clean observations are used in the OLS computation of the initial parameter

estimates. Table 4.9 summarizes the performance of the estimators tested for Example

4.1.

Table 4.9. Estimator performance for Example 4.1. Mod S&M and Mod C&H
are the modified versions using the R&W robust distances for the measure of leverage
with all other components the same. Unusual residual and π-weights indicate if those
measures are significantly different for the 12 outlying cases. Swamped cases refers to
the number of cases out of the clean 48 that have a standardized residual value exceeding
2.5 in absolute value.

Estimator	MSE for clean cases	Unusual Residual?	Unusual π-weights?	Swamped Cases
OLS	2.716	No	NA	7
S&M	3.063	No	No	8
C&H	3.009	No	No	7
Mod S&M	2.969	No	Yes	8
Mod C&H	3.000	No	Yes	7
S	3.384	No	NA	8
LTS	3.267	No	NA	8
CEP1	0.891	Yes	Yes	0
CEP2	0.890	Yes	Yes	0
CEP3	0.976	Yes	Yes	0
CEP4	0.902	Yes	Yes	0

4.8 Performance of the Proposed Compound Estimators

This section evaluates how well the proposed estimators perform beyond the

single example discussed. Example 4.1 clearly demonstrates the ability of the four

proposed estimators to accommodate the outliers in this high-leverage, high-dimension

example while the C&H and S&M estimators do not. The example does not discriminate the performance between these four proposed procedures. This section tests the compound estimators through Monte Carlo simulations to quantify the ability to accommodate outliers.

4.8.1 Proposed Estimators' Area of Coverage

The scenario for Example 4.1 that develops the need for a new compound estimator uses $n = 60$, $k = 6$ with 20% outliers at a leverage magnitude of $5\sigma_x$ in 2 of the 6 regressor variables and a residual magnitude $10\sigma_e$. The proposed estimators effectively downweight the observations while the other compound estimators did not. The Monte Carlo simulation studies in Chapter 3 and Section 4.4 indicate that the leverage and residual magnitudes (and their two-factor interaction) are the most important factors influencing the performance of the tested procedures. Figure 4.1 provides a leverage and residual magnitude sensitivity analysis for the compound estimators. The measure of performance is whether or not the estimator identifies and downweights the planted outliers given the level of leverage (between $\delta_x = 0.0$ and 10.0 σ_x) and residual (between $\delta_R = 3.0$ and 10.0 σ_e) magnitude. Fixed are the number of observations at 60, the number of regressors at 6, the percentage of outliers in the single multiple point cloud at 20%, and the number of regressors out of 6 with high-leverage points at 2.

A method is successful in a single replicate if the average standardized residual value for the 12 outliers is greater than or equal to 2.5. There are 50 replications and if the method is successful in at least 70% of the relicates, then it is deemed successful for

that combination of leverage and residual magnitude. The actual percentage of correct outlier identification is shown in the Appendix B. An estimator successfully accommodates outliers at all combinations of leverage and residual magnitude *above* the line shown in Figure 4.1. The C&H and S&M estimators do not have any power beyond $\delta_L = 4\sigma_x$. At $\delta_R = 15\sigma_e$ the S&M procedure will downweight the 12 outliers because the initial S-estimate has detected the outliers and the parameter estimates do not change much in the remaining stages. The π-weights are still not unusual. If $\delta_L = 10\sigma_x$, the S&M estimator accommodates the 12 outliers when $\delta_R \geq 20\sigma_e$ for the same reason. Figure 4.1 displays how the proposed compound estimators increase the envelope of performance, particularly in δ_L, over the other estimators. CEP1 and CEP2 are preferred over S&M, C&H, CEP3 or CEP4. Although the levels of n, k, density and number of clouds are fixed, the simulation results from both Chapter 3 and Section 4.4 suggest significant increases in the envelope of performance are possible across a variety of factor level combinations.
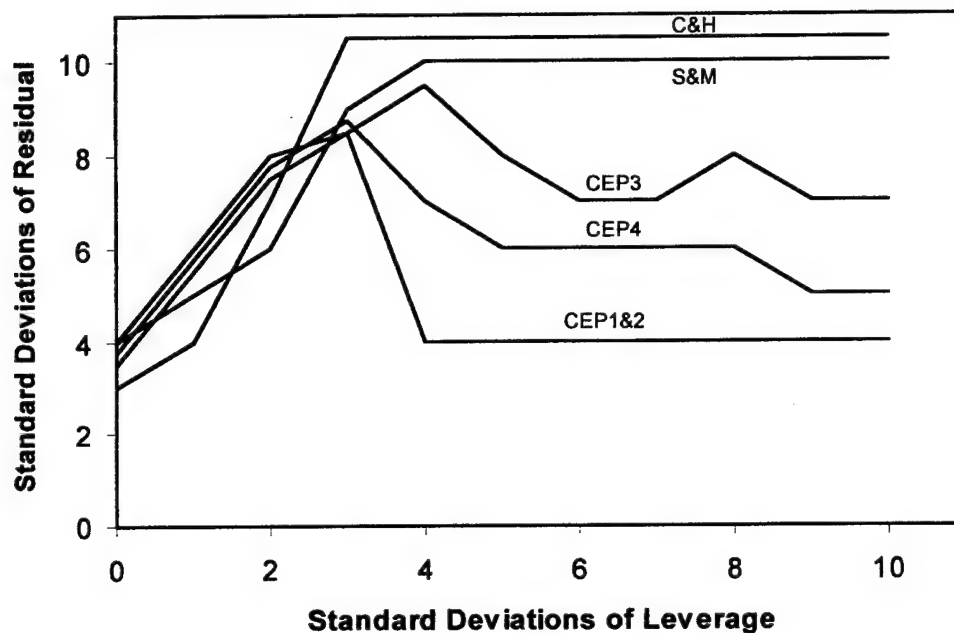
Figure 4.1. Approximate area of coverage for the 6 compound estimators. The data set consists of a single outlying multiple point cloud with factor settings $n = 60$, $k = 6$, outlier density = 20% and outlying in 2 of 6 regressor variables. The X-axis measures the leverage, δ_L, in standard deviation units and the Y-axis measures the cloud's outlying distance in residual, δ_R, in standard deviation units. The area *above* the line for each technique indicates where the estimator is at least 70% effective in identifying the planted outliers. Note that there is no coverage until at least $\delta_R = 15\sigma_e$ for lines S&M and C&H above $\delta_L = 4$ standard deviation units.

4.8.2 Performance in Published Scenarios

Simpson and Montgomery (1998b) conduct a performance study using Monte Carlo simulation in 24 outlier scenarios to evaluate several common and proposed robust regression procedures. The study considers four factors: 1) number of regressors and observations with levels $k=2$, $n=16$; $k=6$, $n=40$; and $k=10$, $n=80$; 2) outlier density with levels 10% and 20%; 3) outlier leverage; and 4) the presence or absence of

approximately 20% high-leverage observations. The regressor variables are placed in a 2 level factorial designed experiment arrangement with levels of ± 1. There are also approximately 20% axial design points with levels of 0 for all but one regressor variable whose level is $\pm\sqrt{k}$. The high-leverage observations replace $\sqrt{k}$ with a value between 5 and 14. Note that the design matrix does not change in the simulation replicates within an outlier scenario. The ith response value is generated by $y_i = \beta' x_i + \varepsilon_i$ where β is the vector of known coefficients for the simulation, x_i is i^{th} row of the design matrix, and ε_i is NID (0, 1) for the clean observations and a large constant for the planted outliers. The measure of performance is the mean square error of estimation defined as

$MSEE = (\hat{\beta}_R - \beta)'(\hat{\beta}_R - \beta)$ where $\hat{\beta}_R$ is the vector of parameter estimates from the robust technique and β is the vector of known model coefficients.

The scenario descriptions and simulation results for 100 replicates of the 24 scenarios are shown in Table 4.10. The average MSEE (AMSEE) is the average of the MSEE for the 100 replicates. The second to last column is the percent of the total observations included in the initial estimate for P1 and the last column is the percent of total observations that P1 should ideally have used in the initial estimate. P1 is an efficient estimator in these scenarios because the ratio of the observed to the expected value is consistently above 95% for the 24 data sets. If P1 uses significantly fewer observations than expected, then many false alarms have resulted. Conversely, if P1 uses more than the expected observations, then it has included high-leverage observations or interior residual outliers in its parameter estimates.

The design matrices for the simulations had to be slightly altered from Simpson and Montgomery (1998b) by adding a realization of $N(0, 0.1^2)$ to every level of the k regressors because the R&W technique and *MM* estimator have difficulty with singularity when the levels are ± 1. This modification does not change the overall results much from Simpson and Montgomery (1998b). Also, for the 6 different design matrices, **X**, used in the 24 outlier scenarios, all measures of leverage for the compound estimators (MVE, *M*-estimates of Covariance, and R&W) correctly assign a large distance to all high-leverage observations and do not assign a large distance to any low-leverage observation. Therefore, the measure of leverage is not a discriminating factor affecting candidate compound estimator performance.

The results for the published estimators in Table 4.10 are consistent with those of Simpson and Montgomery (1998b, page 1044). Of the new proposals, CEP1, CEP3 and CEP4 perform similarly and are moderately better than CEP2. S&M still outperforms all other estimators in these scenarios. CEP1, CEP3 and CEP4 are competitive with S&M except when the 20% high-leverage points are present with another 20% outliers on the interior of X-space (data sets 10-12). The proposed estimators outperform S&M in the high-leverage outlier scenario of data set 17, otherwise the techniques have similar performance. Overall, the proposed estimators are strong performers with only one area of vulnerability (DS10). The performance of the stand-alone *MM* estimator in Table 4.9 should not be overlooked. It is vulnerable only for high-leverage outliers in low

Table 4.10. Average Mean Square Error of Estimation (AMSEE) for robust regression techniques using Simpson and Montgomery (1998b) data sets. CEP1 is the proposed compound estimator using with R&W measures of leverage and P1 initial estimates and CEP2 is the proposed compound estimator with R&W measures of leverage and P2 initial estimate. CEP3 and CEP4 are the proposed estimators using the R&W measures of leverage and 3 or 4 iterations of IRLS in the S&M estimator.

DS	Description	LS	M	LTS	S	MM	C&H	S&M	CE P1	CE P2	CE P3	CE P4	% obs	Exp % obs
1	2V, 12%, int, none	2.171	0.267	0.562	0.461	0.255	0.352	0.274	0.267	0.267	0.264	0.264	84.4	88.0
2	6V, 10%, int, none	2.423	0.222	0.674	0.483	0.211	0.376	0.229	0.220	0.212	0.213	0.212	86.6	90.0
3	10V, 10%, int, none	2.881	0.150	0.532	0.370	0.147	0.275	0.167	0.156	0.150	0.153	0.152	86.3	90.0
4	2V, 19%, int, none	2.438	0.348	0.623	0.405	0.369	0.461	0.336	0.367	0.420	0.380	0.387	77.8	81.0
5	6V, 20%, int, none	4.795	0.238	0.675	0.415	0.240	0.454	0.248	0.243	0.237	0.239	0.239	77.4	80.0
6	10V, 20%, int, none	5.772	0.168	0.511	0.308	0.166	0.314	0.173	0.172	0.168	0.169	0.169	77.2	80.0
7	2V, 12%, int, lev	0.767	0.125	0.349	0.234	0.129	0.230	0.170	0.340	0.399	0.334	0.345	60.7	62.5
8	6V, 10%, int, lev	0.542	0.062	0.436	0.266	0.075	0.348	0.149	0.283	0.280	0.368	0.270	67.5	70.0
9	10V, 10%, int, lev	0.583	0.052	0.321	0.197	0.052	0.248	0.108	0.195	0.192	0.184	0.189	67.3	70.0
10	2V, 19%, int, lev	0.839	0.141	0.310	0.195	0.158	0.233	0.188	0.786	0.897	0.659	0.782	50.6	56.3
11	6V, 20%, int, lev	0.919	0.067	0.388	0.217	0.085	0.322	0.149	0.395	0.537	0.324	0.324	57.0	60.0
12	10V, 20%, int, lev	1.152	0.054	0.292	0.142	0.056	0.237	0.107	0.239	0.234	0.317	0.227	58.2	60.0
13	2V, 12%, ext, lev	1.742	0.713	0.433	0.334	0.238	0.325	0.225	0.253	0.263	0.250	0.252	71.8	75.0
14	6V, 10%, ext, lev	1.598	1.127	0.757	0.514	0.309	0.624	0.297	0.236	0.235	0.231	0.229	76.8	80.0
15	10V, 10%, ext, lev	2.872	0.265	0.472	0.291	0.098	0.374	0.144	0.171	0.167	0.162	0.164	76.5	80.0
16	2V, 19%, ext, lev	2.183	1.469	0.533	0.397	0.779	0.430	0.285	0.256	0.264	0.253	0.254	71.8	75.0
17	6V, 20%, ext, lev	2.978	3.259	1.283	1.005	0.936	1.096	0.725	0.234	0.233	0.230	0.227	76.8	80.0
18	10V, 20%, ext, lev	5.706	7.027	0.510	0.337	0.160	0.416	0.169	0.169	0.166	0.161	0.161	76.5	80.0
19	2V, 12%, int/ext	1.347	0.310	0.352	0.234	0.129	0.255	0.167	0.317	0.332	0.286	0.292	66.3	68.8
20	6V, 10%, int/ext	1.130	0.606	0.565	0.403	0.167	0.467	0.217	0.256	0.253	0.246	0.248	72.3	75.0
21	10V, 10%, int/ext	1.786	0.117	0.396	0.238	0.079	0.316	0.122	0.175	0.173	0.170	0.173	72.0	75.0
22	2V, 19%, int/ext	1.884	1.279	0.418	0.294	0.484	0.334	0.251	0.386	0.396	0.381	0.394	66.0	68.8
23	6V, 20%, int/ext	2.040	1.508	0.553	0.385	0.390	0.467	0.278	0.285	0.281	0.277	0.279	67.5	70.0
24	10V, 20%, int/ext	3.512	1.043	0.404	0.231	0.130	0.324	0.147	0.193	0.191	0.185	0.188	67.3	70.0
Sum		54.06	20.62	12.35	8.36	5.84	9.28	5.33	6.59	6.94	6.44	6.42		

dimension. For this reason, CEP1, which incorporates an *MM* estimator, is probably the best alternative to protect against multiple outliers in high-dimension. Although CEP1 is computationally complex, a data set of $n = 100$ observations with $k = 10$ regressor variables requires approximately 20 seconds on a modest PC (2 seconds for S&M and 5 seconds for C&H). The procedure could be made much more computationally efficient if it did not go through the *S-Plus* interface.

4.9 Summary

This chapter develops new compound estimators that greatly expand the region of effective performance in the presence of high-leverage outliers over existing procedures. A comprehensive simulation study on common measures of leverage indicates that the R&W procedure is the most robust across a variety of scenarios. The improved measure of leverage alone does not significantly improve a compound estimator's performance with high-leverage outliers unless several more iterations are added to the IRLS solution to the *GM* normal equations. The common high-breakdown initial estimators (LMS, LTS and *S*) are vulnerable to the high-leverage outliers and provide inferior initial estimates. Good initial estimates are essential because often the final estimates from a compound estimator do not significantly change when they should. Also, the estimate of scale for a compound estimator is based on the residuals from the initial estimate.

Our approach is to provide an initial estimate based only on observations that are not high-leverage points or residual outliers. This provides an efficient estimator because 50% of the observations are not removed from the sample as they are in LMS and LTS.

The R&W and *MM* filters remove a variable percentage of the data. Studies show that this scheme leaves approximately 95% of the clean observations in the initial estimate independent of outlier density or dimension. Therefore, the proposed initial estimator is efficient, high-breakdown and bounded-influence. The next stages of a compound estimator can then smooth in the outliers based on the user's downweighting philosophy through the choice of ψ-function.

Simulation studies indicate our proposed estimator CEP1 is competitive with the top performing robust regression techniques tested in published scenarios and preferred in high-leverage, high-dimension scenarios. CEP1 uses the R&W robust distances for the leverage measure, an initial estimate from OLS after the R&W and *MM* filters, and an LMS estimate of scale based on the residuals from the initial estimate. Figure 4.1 provides an indication of the substantial improvement in performance this estimator has in the presence of high-leverage outliers over other robust regression techniques.

Chapter 5

Resampling Methods for Variable Selection in

Least Squares and Robust Regression

5.1 Introduction

An important aspect of the regression model building strategy is selecting the appropriate subset from the candidate regressor variables. We consider the usual multiple regression model, $y = X\beta + \varepsilon$ where X is the $n \times p$ matrix of regressor variables, β the p vector of parameters and ε the random error assumed to be independent and identically distributed (i.i.d.) with mean 0 and variance $\sigma^2 I$. The β vector is partitioned into an active variable set, β_1 of $p - q$ parameters and inactive set β_2 of q parameters to test the hypothesis that

$$H_O: \beta_2 = 0$$
$$H_A: \beta_2 \neq 0.$$

Failure to reject the null hypothesis suggests there is no evidence that any of the regressor variables in set β_2 have an effect on the response value.

The goal of a variable selection procedure is to have the significant regressor variables included in set β_1 with high probability, while simultaneously achieving a high probability that the insignificant variables are contained in set β_2. The regression model building strategy is an iterative process that involves selection of an active subset of the p regressors followed by model diagnostics to assess the fit. The objective is to find the

best subset of the p parameters to include in the model that leads to good prediction capability yet minimizes the variance of prediction. The first objective would suggest including all p variables while the second suggests using as small of a subset as possible because the variance of prediction always increases as regressor variables are added. Models with fewer variables are also preferred for simplicity in interpretation and ease of future data collection.

There are numerous variable selection methods available to the analyst. The simplest approach is to retain only the variables whose ratio of coefficient to the standard error is significant. This t-test approach is not reliable with increasing dimension, particularly when dependencies between regressor variables exist. A common alternative is the class of computer-intensive variable selection methods (e.g. forward, backward, stepwise, and best subsets regression). The selection criteria are often based on F-tests (F-to-enter and F-to-leave) or Mallows's (1973) C_p criterion;

$C_p = \dfrac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 - n + 2p$ where $\hat{y}_i$ is the predicted value and $\hat{\sigma}^2$ is typically the

mean square error (MS_E) from the full model. Unfortunately, Miller (1990) demonstrates that the F tests and Mallow's C_p criterion are poor for model selection as are the R^2 and adjusted R^2 measures. Breiman (1995) states that the preferred measure of performance for variable selection in regression is some measure of prediction error.

The resubstitution or apparent prediction error for a regression model is defined by $\hat{\Delta}_{App} = n^{-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Note that this quantity differs from the usual MS_E estimate because n is used in place of $n - p$. The final prediction error (FPE) criteria (Zhang,

1992) accounts for the number of regressors in the model and is computed by $n\hat{\Delta}_{App} + \lambda p$ where λ is the penalty constant for including extra variables. When $\lambda = 2$, FPE can be shown to be equivalent to Mallows's C_p. It has been well documented that the FPE, C_p and $\hat{\Delta}_{App}$ measures are highly biased (Miller, 1990, Zhang, 1992, Shao, 1993, and Breiman, 1995, Davison and Hinkley, 1997) and not recommended for variable selection. Direct minimization of these measures leads to models that have too many significant variables; the dimension of β_1 is too large.

Several authors have proposed computationally complex resampling methods to address the shortcomings of the usual methods for variable selection. Common resampling methods are cross-validation and bootstrapping. We describe several cross-validation and bootstrap resampling methods to calculate prediction error in Section 5.2. Each of these methods suggests selecting the model with the minimum prediction error among the competitors. Our approach is to relax the requirement for the absolute minimum prediction error and select a model that has the fewest number of variables and a low (not necessarily minimum) prediction error. This criterion is effective with both cross-validation and bootstrap estimates of prediction error. It is our belief that in low dimension ($p < 10$), a reasonable strategy is to look at a screeplot (scatterplot of the number of parameters versus prediction error) of candidate models of increasing dimension. The model with the fewest parameters where the curve levels off is selected. For example, the screeplot in Figure 5.1 suggests that although the 7-parameter model has the minimum value of prediction error, little improvement is gained after five parameters are included in the model. In Section 5.3 we describe a simulation

experiment that tests several resampling prediction error methods on a published data set using both model selection criteria: the absolute minimum prediction error and our recommended strategy. The results in Section 5.4 indicate that the absolute minimum criterion is not required and effective model selection is possible with the proposed heuristic. Section 5.5 extends the simulation to higher dimension and also evaluates performance when the signal-to-noise ratio is not high. Section 5.6 explores the usefulness of the various resampling model selection schemes in the presence of outlying observations using a robust regression estimator. Recommendations and conclusions are offered in Section 5.7
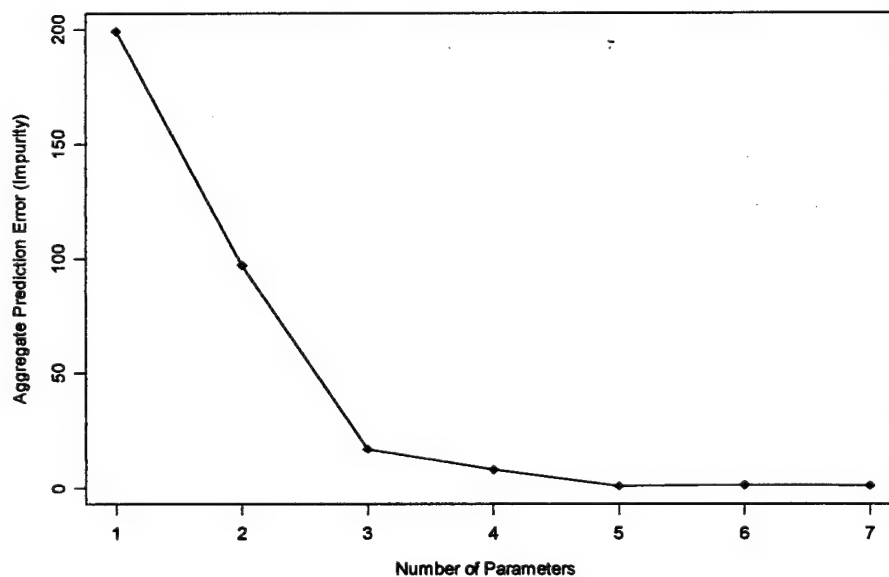
Figure 5.1. Representative screeplot of aggregate prediction error.

5.2 Resampling Measures of Prediction Error

The two classes of resampling methods currently recommended to calculate a measure of prediction error for variable selection are cross-validation and bootstrapping. Cross-validation procedures partition the data into two disjoint sets. The model is fit with one set (the training set) and it is subsequently used to predict the responses for the observations in the second set (assessment set). Bootstrap procedures form many samples of the original data by resampling with replacement. Details of the methods and their application to the variable selection problem in regression are outlined below.

5.2.1 Cross-Validation Procedures

An intuitively appealing method to calculate a predicted response value is to use the parameter estimates from the fit obtained by omitting the observation to be predicted. This predicted response value is denoted by $\hat{y}_{(i)}$. Then $\hat{\Delta}_{CV,1} = n^{-1}\sum_{i=1}^{n}(y_i - \hat{y}_{(i)})^2$ is computed as the leave-one-out cross-validation estimate of average prediction error for a model. Apart from the n^{-1} term, this quantity is the predicted error sum of squares (PRESS) statistic in least squares (Allen, 1971). The PRESS statistic can be calculated as

$$\sum_{i=1}^{n}\left(\frac{e_i}{1 - h_{ii}}\right)^2$$ where $h_{ii} = \mathbf{x}_i(\mathbf{X'X})^{-1}\mathbf{x}_i'$. Note that least squares does not require n separate fits

for PRESS. Other regression estimators (e.g. robust) do require all n fits for the leave-one-out cross-validation estimate of prediction error. Shao (1993) proves with asymptotic results and simulations that the model with the minimum PRESS statistic or

leave-one-out cross-validation estimate of prediction error is often overspecified. He recommends using K-fold cross-validation that leaves a subset of observations out.

Quenouille (1949) explored the idea of leaving two observations out of the training set and Stone (1974) extended the method to more than two. In K-fold cross-validation, the training set omits approximately n/K observations from the training set rather than a single observation like PRESS. To predict the response values for the kth assessment set, $S_{k,a}$, all observations apart from those in set k are in the training set, $S_{k,t}$, and these are used to estimate the model parameters. The K-fold cross-validation average prediction error is $\hat{\Delta}_{CV,K} = n^{-1} \sum_{i=1}^{n} \left(y_i - \hat{y}_{(k,i)} \right)^2$ where $\hat{y}_{(k,i)}$ is the predicted response for observation i belonging in assessment set $S_{k,a}$.

One approach to the K-fold cross-validation estimate of prediction error is to randomly select the n/K observations to form the assessment set. This process is repeated numerous times and the prediction errors are averaged. Breiman et al. (1984) propose a less computationally intense scheme that randomly partitions the data into K different disjoint sets. Davison and Hinkley (1997) recommend $K = min\ (n^{1/2}, 10)$ in practice. This procedure decreases the variance of prediction error over that of the leave-one-out cross-validation estimate but at the expense of increased bias. Surprisingly, Shao (1993) demonstrates that the smaller the training set, the better the K-fold estimate is for model selection.

To reduce the bias, Burman (1990) recommends the adjusted K-fold cross-validation estimate of prediction error as

$$\hat{\Delta}_{ACV,K} = \hat{\Delta}_{CV,K} + \hat{\Delta}_{App} - \sum_{k=1}^{K} p_k \left(n^{-1} \sum_{i=1}^{n} (y_i - \hat{y}_{(k,i)})^2 \right)$$ where p_k is the ratio of observations

in assessment set k to the total n and $\hat{y}_{(k,i)}$ is the predicted response for the ith observation

from the fit with training set $S_{t,k}$. The Breiman and Spector (1992) simulations

demonstrate that the performance of the adjusted cross-validation prediction error

estimate is slightly worse than the standard biased K-fold cross-validation prediction

error for least squares variable selection. Shao (1993) shows that both the leave-one-out

and K-fold cross-validation procedures have a negligible probability of selecting an

underspecified model. The challenge is avoiding an overfit model.

5.2.2 Bootstrap Procedures

Bootstrap estimators in regression have received considerable attention in the

literature since their introduction by Efron (1979). Wu (1986) provides the theoretical

results for bootstrap methods applied to regression. Hall (1989) proves that inference

procedures in regression, such as confidence intervals, based on the bootstrap estimate

are more accurate than standard inference procedures even if the error is Gaussian.

The fundamental element of a bootstrap procedure is the bootstrap sample. For

bootstrapping pairs in regression (Efron, 1982), the sample is formed by randomly

sampling with replacement n times both a response value and its associated vector of

regressor variable values from the original sample. The bootstrap sample may contain an

observation from the original sample once, multiple times or not at all. In fact, the

probability that an observation is included in a bootstrap sample of size n is $1 - e^{-1} =$

0.632 (Efron and Tibshirani, 1997). A regression model is then fit to the bootstrap sample to obtain the bootstrap parameter estimates $\hat{\beta}^*$. A large number of bootstrap samples ($B \geq 100$, Davison and Hinkley, 1997) are constructed from the original sample for model inference.

For the variable selection problem, the estimate of the average prediction error for the bth bootstrap sample is $\hat{\Delta}_b = n^{-1} \sum_{i=1}^{n} (y_i - \mathbf{x}_i \hat{\beta}_b^*)^2$ where y_i and $\mathbf{x}_i$ are from the original sample. Efron (1983) provides the unbiased estimator of prediction error for the bth sample as $\hat{\Delta}_{b,unbiased} = n^{-1} \sum_{i=1}^{n} (y_i - \mathbf{x}_i \hat{\beta})^2 + n^{-1} \sum_{i=1}^{n} (y_i - \mathbf{x}_i \hat{\beta}_b^*)^2 - n^{-1} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^* \hat{\beta}_b^*)^2$ where $\mathbf{x}_i^*$ is the vector of regressor values for the ith observation in the bth bootstrap sample. The overall unbiased bootstrap estimate of average prediction error is simply

$$\hat{\Delta}_{BS} = B^{-1} \sum_{b=1}^{B} \hat{\Delta}_{b,unbiased}$$ where B is the number of bootstrap samples. Shao (1996) shows that selecting the model with the minimum $\hat{\Delta}_{BS}$ is inconsistent. Inconsistency implies that the probability the true model has the minimum bootstrap average prediction error does not equal 1.0 as n approaches infinity. Shao corrects this inconsistency for bootstrapping pairs by using substantially fewer than n observations to construct the bootstrap samples. This procedure uses the biased estimate of prediction error. Breiman (1996), motivated by the 0.632 probability that an observation is selected in a bootstrap sample, notes that using bootstrap samples of size $2n$ has little effect on OLS variable selection.

5.2.3 Other Modifications to Resampling Methods for Variable Selection

Breiman and Spector (1992) explore the use of cost admissibility (penalty for adding variables) with bootstrap and cross-validation prediction error for variable selection. Their empirical results indicate that this modification only slightly increases the probability of selecting the correct model. This is an important result because most resampling estimates of prediction error do not account for the number of variables in the model.

Breiman (1992) recommends the little bootstrap estimate of prediction error for variable selection in linear models. The prediction error for a k variable model using this approach is $\hat{\Delta}_{App}(k) + 2B_t(k)$. The little bootstrap error, $B_t(k)$, is the resubstitution error from the model selected using $\mathbf{y}^* = \mathbf{y} + \widetilde{\varepsilon}$ where $\widetilde{\varepsilon}$ is the vector of variates from NID (0, $t^2\sigma^2$) with $0.6 < t < 0.8$. The MS_E for the full model is used as an estimate of σ^2. Breiman shows that the little bootstrap is unbiased and superior to C_p, F-to-enter, and F-to-leave for variable selection for fixed designs.

Breiman (1996) suggests *bagging* (bootstrap aggregating) regressor variables. For each of the B samples formed by bootstrapping pairs, perform a forward selection to obtain a 1 variable model, 2 variable model, ... k variable model. The $n \times k$ matrices of predicted values from these k models are averaged across the B bootstrap samples. The model with the lowest average prediction error is selected. Limited simulation results indicate that this procedure performs better than standard forward selection. It is unclear how to proceed if the same variables are not consistently selected in the B samples for a given dimension.

Davison and Hinkley (1997) describe a hybrid estimate of bootstrap prediction error for variable selection adapted from Efron and Tibshirani (1997). The hybrid estimate of prediction error weights the apparent error and the bootstrap cross-validation (BCV) error. The BCV error is calculated from the predicted and observed values of those observations not included in the bootstrap sample. The recommended weights from theory and practice are 0.632 for the BCV error and (1-0.632) for the apparent error. The authors' empirical evidence suggests that this procedure is superior.

5.3 An Alternative Criterion for Variable Selection

Recent results indicate that many of the classical measures used for variable selection such as R^2, adjusted R^2, C_p, and PRESS, are highly biased and not suitable for variable selection. The computer selection methods of forward, backwards, and stepwise are based on these measures and also provide biased results. Many of the arguments against these procedures are derived from asymptotic properties and assume that the candidate model with the minimum (or maximum for R^2 measures) value of the statistic is selected. We believe satisfactory results, and in many cases superior results, are possible by relaxing the requirement of selection by the minimum value of the statistic. Rather, one would select the model that has a low prediction error with the fewest variables. This procedure is a more realistic representation of what a practitioner is likely to do given the prediction errors from the candidate models. Obviously, there is more subjectivity with this criterion than simply selecting the model with the minimum prediction error.

To illustrate the methodology, the average prediction errors ($\hat{\Delta}_{BS}$) from 100 bootstrap samples each for models with an increasing number of active variables are displayed in Table 5.1. The model is $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where $\mathbf{X}$ is the design matrix formed by augmenting the four regressor variables from the Gunst and Mason (1980) data with a column of ones, $\boldsymbol{\beta}$ is the known vector of parameters and $\boldsymbol{\varepsilon}$ is the vector of NID ($\mathbf{0}, \sigma^2\mathbf{I}$) error terms. This data set (shown in the *S-Plus* code in Appendix C) is used extensively in the Shao (1993 and 1996) studies and in sections 5.4 through 5.6. The column headings of Table 5.1 display the known generating vector $\boldsymbol{\beta}$ used to calculate the response values. Our procedure looks at the change in prediction error going from a model of dimension j to dimension $j + 1$. If there is only a slight decrease, then the smaller dimension model is preferred. In the second column of Table 5.1, our strategy would correctly choose the 2-parameter model (intercept and β_3) rather than the model with the minimum prediction error; the 5-parameter model. Similarly, the proposed method would select the correct model (the shaded cells) for the other columns.

Table 5.1. Average prediction error from 100 bootstrap samples as a function of the number of variables in the model. The column headings are the true model.

p	[2, 0, 0, 4, 0]	[2, 0, 0, 4, 8]	[2, 9, 0, 4, 8]	[2, 9, 6, 4, 8]
1	18.51	130.03	188.90	266.01
2	0.96	15.75	22.07	22.27
3	1.01	0.89	3.36	9.35
4	1.05	0.95	0.95	4.02
5	0.95	0.96	0.93	0.96

In practice, the proposed criterion requires subjective judgment. For simulation studies, we must specify the minimum change in prediction error required to select the next higher dimension model. We follow the impurity logic used to split and terminate nodes in Classification and Regression Trees (Breiman et al., 1984). If the change in prediction error does not exceed a certain percentage of the prediction error for the null model (intercept only), then the lower dimension model is selected. For example, if our minimum change in prediction error criterion were 1%, then the difference in prediction error must be at least 0.185 between models of size j and $j + 1$ for the first column in Table 5.1. We are not advocating using a specific percentage as much as carefully inspecting the prediction errors between candidate models. The percentages are useful for comparative studies in simulations.

5.4 A Simulation Study

The simulation scenarios reported in Shao (1996) provide an ideal test bed for the proposed change in prediction error criterion. The regressor variables are those from the Gunst and Mason (1980) data set with $n = 40$ cases and responses generated as described in Section 5.3. Some of the constants in β are 0; therefore, the objective of the study is to assess several resampling methods' ability to correctly identify the active set of regressor variables using both the minimum prediction error and the proposed change in prediction error criteria. Shao's objective is to demonstrate that using a much smaller bootstrap sample size than n leads to consistent variable selection for the minimum prediction error criterion while all other techniques (PRESS, C_p, and the bootstrap with full sample) have

considerably less capability to select the proper model. A useful outcome of our study would be to demonstrate that these inconsistent, yet commonly used, techniques can provide reliable model selection if we change the criterion.

5.4.1 Simulation Details

The 1000 simulation replicates generate the data sets exactly as in Shao (1996). That is, the Gunst and Mason set of regressor variables and the response values calculated by specifying the known parameters and adding a vector of standard normal variates to $X\beta$. The measure of effectiveness for a procedure is the proportion of the 1000 replicates that the correct model of known dimension j is selected. The resampling estimates of prediction error used are the leave-one-out cross-validation estimate, the K-fold cross-validation, the adjusted K-fold cross-validation, the bias-corrected bootstrap of using sample size n, and the bootstrap with sample size $n/2$. Following the Davison and Hinkley (1997) recommendations, the value of K is 6 and the number of bootstrap samples, B, is fixed at 100 per replication. The prediction errors for these 100 bootstrap samples are averaged and then the model selection criterion (minimum prediction error or change in prediction error) is applied. For the change in prediction error, we run pilot studies to find reasonable values for the constant defined as the percentage of null model prediction error. We follow Shao's suggestion that it is not practical to evaluate all $\binom{p}{2}$ possible models and also evaluate one model in each dimension.

5.4.2 Simulation Results

All simulation results are summarized in tables. The known correct model is shaded in the tables. The proportion of simulation replicates that the minimum prediction error criterion selects the model is the first entry in each cell. The proportion from the proposed criterion is the second entry in each cell.

Table 5.2 displays the results for the model with the intercept and β_3 active. All five procedures have above a 98% chance of correctly identifying the true model with the proposed selection criterion. This suggests that a practitioner comparing models with the PRESS statistic would likely have made the correct choice upon careful examination of the prediction errors. Consistent with the results reported by Shao (1996), the minimum prediction error criterion has difficulty with overfitting for most methods. No prediction error method except the bootstrap half sample reliably selects the correct model under the minimum prediction error criterion. A surprising result is that only 50.9% of the 100,000 bootstrap samples (100 bootstrap samples x 1000 replicates) selected the correct model with the minimum prediction error criteria. Yet, when prediction error is averaged over the 100 bootstrap samples, the correct model is selected in approximately 95% of the 1000 replicates.

Table 5.2. Results for 1000 simulation replicates for model selection with 2 active parameters in the Shao (1996) data sets. The selection percentages for the true model [2, 0, 0, 4, 0] are shaded. The top number in each cell is the proportion of replications that the model was selected using the minimization of prediction error criterion and the bottom uses the proposed change in prediction error criterion with constant .025. The values in brackets are the proportion of 100,000 bootstrap samples that the model is selected. Results are accurate to approximately ±0.03.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.001]	0.000 [0.018]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.005]
β_0, β_3	0.687	0.648	0.633	0.781 [0.414]	0.948 [0.509]
	0.991	0.980	0.984	0.998 [0.939]	0.999 [0.847]
$\beta_0, \beta_3, \beta_4$	0.164	0.188	0.183	0.122 [0.210]	0.045 [0.214]
	0.002	0.004	0.002	0.001 [0.014]	0.001 [0.029]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.080	0.076	0.084	0.060 [0.161]	0.005 [0.138]
	0.001	0.001	0.001	0.000 [0.017]	0.000 [0.045]
$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	0.069	0.088	0.100	0.037 [0.214]	0.002 [0.122]
	0.006	0.015	0.013	0.001 [0.029]	0.000 [0.074]

The results in Tables 5.3 and 5.4 for the 3 and 4 active parameter models, respectively, further confirm the superiority of the change in prediction error criterion. There is virtually a 100% chance of selecting the correct model with the proposed criterion independent of the resampling procedure choice. Contrary to most other published results, the leave-one-out cross-validation estimate slightly outperforms the K-fold and adjusted K-fold methods for the minimum prediction error criteria. Also, the adjusted K-fold method is slightly worse than the K-fold which agrees with Breiman and Spector (1992). The results in Table 5.5 with all 5 parameters active favor the minimum prediction error criterion. This is not unexpected because this criterion rarely selects an underspecified model.

Table 5.3. Results for 1000 simulation replicates for model selection with 3 active parameters in the Shao (1996) data sets. The selection percentages for the true model [2, 0, 0, 4, 8] are shaded. The top number in each cell is the proportion of replications that the model was selected using the minimization of prediction error metric and the bottom uses the change in prediction error criteria with constant .025. The values in brackets are the proportion of 100,000 bootstrap samples that the model is selected. Results are accurate to approximately ±0.03.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.003]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.000]
β_0, β_3	0.000	0.000	0.000	0.000 [0.008]	0.000 [0.000]
	0.000	0.000	0.000	0.000 [0.058]	0.000 [0.000]
$\beta_0, \beta_3, \beta_4$	0.728	0.673	0.671	0.783 [0.462]	0.959 [0.561]
	1.000	1.000	1.000	1.000 [0.941]	1.000 [0.979]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.180	0.232	0.227	0.144 [0.252]	0.032 [0.236]
	0.000	0.000	0.000	0.000 [0.001]	0.000 [0.010]
$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	0.092	0.227	0.102	0.073 [0.278]	0.009 [0.200]
	0.000	0.000	0.000	0.000 [0.001]	0.000 [0.011]

Table 5.4. Results for 1000 simulation replicates for model selection with 4 active parameters in the Shao (1996) data sets. The selection percentages for the true model [2, 9, 0, 4, 8] are shaded. The top number in each cell is the fraction of replications that the model was selected using the minimization of prediction error criterion and the bottom uses the change in prediction error criterion with constant .025. The values in brackets are the proportion of 100,000 bootstrap samples that the model is selected. Results are accurate to approximately ±0.03.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val K= 6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.002]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.000]
β_0, β_3	0.000	0.000	0.000	0.000 [0.014]	0.000 [0.000]
	0.000	0.000	0.000	0.000 [0.003]	0.000 [0.000]
$\beta_0, \beta_3, \beta_4$	0.000	0.000	0.000	0.000 [0.005]	0.000 [0.019]
	0.000	0.000	0.000	0.000 [0.019]	0.000 [0.018]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.805	0.781	0.772	0.833 [0.549]	0.958 [0.616]
	0.997	1.000	1.000	1.000 [0.957]	0.999 [0.911]
$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	0.195	0.219	0.228	0.167 [0.433]	0.042 [0.363]
	0.003	0.000	0.000	0.000 [0.021]	0.001 [0.071]

Table 5.5. Table 5.3. Results for 1000 simulation replicates for model selection with 3 active parameters in the Shao (1996) data sets. The selection percentages for the true model [2, 9, 6, 4, 8] are shaded. The top number in each cell is the proportion of replications that the model was selected using the minimization of prediction error criterion and the bottom uses the change in prediction error criterion with constant .025. The values in brackets are the proportion of 100,000 bootstrap samples that the model is selected. Results are accurate to approximately ±0.03

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross–Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.001]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.000]
β_0, β_3	0.000	0.000	0.000	0.000 [0.010]	0.000 [0.000]
	0.000	0.000	0.000	0.000 [0.001]	0.000 [0.000]
$\beta_0, \beta_3, \beta_4$	0.000	0.000	0.000	0.000 [0.002]	0.000 [0.002]
	0.000	0.000	0.000	0.000 [0.005]	0.000 [0.001]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.002	0.000	0.000	0.000 [0.027]	0.000 [0.067]
	0.049	0.041	0.038	0.019 [0.156]	0.020 [0.197]
$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	0.998	1.000	1.000	1.000 [0.960]	1.000 [0.930]
	0.951	0.959	0.962	0.981 [0.838]	0.980 [0.802]

5.5 Extensions to Noisy and High-Dimension Data Sets

The minimum change in prediction error is the superior criterion for any resampling method in the data sets used in Shao (1996). The success of the procedure may be attributed to the low dimension of the data (4 regressor variables), the high signal-to-noise ratio or possibly a combination of both. We conduct some substudies in this section to further characterize the performance of both variable selection criteria.

5.5.1 High-Dimension Data

We modify the Gunst and Mason data set of regressor variables to include 5 additional variables whose levels are generated from a NID $(0, 0.7^2)$ distribution. This approximately matches the current levels for the majority of the original 4 regressors.

The five additional variables have no effect on the generated response values as the β' vector is now [2, 9, 6, 4, 8, 0, 0, 0, 0, 0]. The response variable is still generated as $y = X\beta + \varepsilon$ where ε is NID $(0, \sigma^2 I)$ with $\sigma^2 = 1$.

The probabilities in Table 5.6 again indicate that the change in prediction error criterion performs better than the minimum prediction error criterion. The minimum prediction error criterion overfits models except with the bootstrap resampling method using half samples. Note that in only 42.8% of the 100,000 bootstrap samples was the correct model selected under this criterion for the bootstrap half sample method. In contrast, the proposed criterion selects the correct model in over 80% of the bootstrap samples using the full sample. The change in prediction criterion produces the following important findings: 1) the bootstrap using the full sample is best, 2) the leave-one-out cross-validation outperforms the K-fold cross-validation procedures, 3) the bias adjusted K-fold is slightly preferred to the ordinary K-fold estimate of prediction error, and 4) any resampling method has a high probability of selecting the correct model.

Table 5.6. Results for 1000 simulations for model selection from Shao (1996). The first four regressors are the Gunst and Mason data and the last five regressors are variates from NID $(0,0.7^2)$ and ε is NID $(0,\sigma^2 I)$. The true model, $\beta' = [2, 9, 6, 4, 8, 0, 0, 0, 0, 0]$, is shaded. The top number in each cell is the proportion of replicates the model was selected using the minimization of prediction error criterion and the bottom uses the change in prediction error criterion with constant .001. The values in brackets are the proportion of 100,000 bootstrap samples that the model is selected. Results are accurate to approximately ±0.03.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val = 6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000 0.000	0.000 0.000	0.000 0.000	0.000 [0.000] 0.000 [0.000]	0.000 [0.001] 0.000 [0.000]
β_0, β_3	0.000 0.000	0.000 0.000	0.000 0.000	0.000 [0.001] 0.000 [0.000]	0.000 [0.000] 0.000 [0.000]
$\beta_0, \beta_3, \beta_4$	0.000 0.000	0.000 0.000	0.000 0.000	0.000 [0.000] 0.000 [0.000]	0.000 [0.002] 0.000 [0.000]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.000 0.000	0.000 0.000	0.000 0.000	0.000 [0.000] 0.000 [0.000]	0.000 [0.001] 0.000 [0.000]
$\beta_0 - \beta_4$	0.666 0.950	0.656 0.915	0.637 0.932	0.684 [0.350] 0.977 [0.808]	0.892 [0.428] 0.963 [0.539]
$\beta_0 - \beta_5$	0.113 0.006	0.175 0.014	0.126 0.014	0.116 [0.163] 0.003 [0.022]	0.080 [0.187] 0.008 [0.041]
$\beta_0 - \beta_6$	0.070 0.011	0.074 0.018	0.080 0.016	0.064 [0.119] 0.004 [0.029]	0.022 [0.128] 0.006 [0.057]
$\beta_0 - \beta_7$	0.065 0.010	0.052 0.018	0.055 0.009	0.050 [0.105] 0.005 [0.038]	0.006 [0.097] 0.010 [0.082]
$\beta_0 - \beta_8$	0.051 0.016	0.052 0.020	0.057 0.018	0.055 [0.107] 0.011 [0.048]	0.000 [0.079] 0.007 [0.118]
$\beta_0 - \beta_9$	0.035 0.007	0.041 0.015	0.045 0.011	0.031 [0.155] 0.000 [0.055]	0.000 [0.076] 0.006 [0.163]

5.5.2 High-Dimension and Noisy Data

All of the previous correctly specified models (the shaded models in Tables 5.2 – 5.6) are highly significant. That is, the signal-to-noise ratio is high as evidenced by the R^2 values ranging from 0.98 and 0.995. We would typically not expect to see such high R^2 values in practice. Also, there is a peculiarity in the Gunst and Mason data. Section 5.6 details that there are 12 observations out of the total 40 in extreme X-space (high-

leverage). For these reasons, we temporarily abandon the Gunst and Mason data. This sub-study evaluates the performance of the resampling algorithms and the two model selection criteria when the signal-to-noise ratio and remoteness in X-space are not as extreme.

The artificial data set generates standard normal variates for the design matrix, $\mathbf{X}$, of dimension $n = 40$ observations and $k = 9$ regressor variables augmented with a column of ones. For this study, the design matrix changes for each of the 1000 replications to represent the X-random, as opposed to X-fixed, case for regression. Breiman and Spector (1992) state that significant differences exist between the two assumptions with respect to variable selection and the X-random designs are appropriate for most analysis. The response variable is generated as usual, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ except that $\boldsymbol{\varepsilon}$ is NID $(\mathbf{0}, \sigma^2 \mathbf{I})$ with $\sigma = 10$. The known vector of parameters is the same as the previous experiment, $\boldsymbol{\beta}' = [2, 9, 6, 4, 8, 0, 0, 0, 0, 0]$. The R^2 values with the new distribution of the error term and design matrix range between 0.65 and 0.75. This amount of noise in the data is more realistic for many applications.

The minimum prediction error criterion (the values on top of each cell in Table 5.7) have similar results to the change in prediction error criterion with constant equal to 0.001 (middle values in each cell). Both criteria do not reliably identify the correct model using cross-validation or the bootstrap with the full sample size. The only procedure that does not consistently lead to overfit models is the bootstrap using a sample size of $n = 20$.

One possible solution to make these resampling procedures more reliable is to increase the constant from 0.001. This constant value suggests that the models of the next higher dimension will be selected if the change in prediction error exceeds 1/10 of 1% of the prediction error from the null model. The middle value of each cell in Table 5.7 indicates that we rarely select models that are underfit. Assumption of more risk of underfitting by increasing the value of the constant can lead to a higher probability of correct model selection. The last value in each cell of Table 5.7 is the proportion of replications that the change in prediction error criterion selects the model if the constant is changed to 0.03. The constant, calculated from pilot studies, is set to achieve a balance between underfit and overfit models. The best prescription still appears to be the bootstrap with the half sample size for either criterion. However, the cross-validation and bootstrap with the full sample are competitive if the change in prediction error criterion is used.

Note that the proportions in Tables 5.6 and 5.7 for the proposed change in prediction error criterion are conservative for the correct (shaded) model. The programming logic does not adequately address the situations when there are significant drops in prediction error in higher dimension models but the prediction error is still greater than the correctly specified model. To illustrate, consider the following vector of average prediction errors $\hat{\Delta} = [1000, 500, 100, 50, 15, 14, 45, 60, 25, 30]$. The correct model is the 5-parameter model with prediction error 15. The change in prediction error criterion with constant 0.03 as programmed selects the 9-parameter model because the

prediction error has decreased by more than 30 (0.03 * 1000). It is difficult to capture the

subjective nature of the process; however, the logic errs to the conservative side.

Table 5.7. Results for 1000 simulation replicates for model selection. All regressor variable values are generated from a standard normal distribution. The response is generated from the vector $\beta' = [2, 9, 6, 4, 8, 0, 0, 0, 0, 0]$ with ε NID $(0, \sigma^2 I)$ and $\sigma = 10$. The top value in each cell is the proportion of the 1000 replicates the model was selected using the minimization of prediction error criterion, the middle value is the change in prediction error criterion with constant 0.001, and the bottom value is the change in prediction error criterion with constant 0.03. The values in brackets are the proportion of 100,000 bootstrap samples that the model is selected.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.013]	0.000 [0.039]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.000]
	0.000	0.003	0.003	0.000 [0.001]	0.002 [0.001]
β_0, β_3	0.000	0.000	0.000	0.000 [0.003]	0.000 [0.000]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.000]
	0.000	0.002	0.001	0.000 [0.008]	0.001 [0.002]
$\beta_0, \beta_3, \beta_4$	0.004	0.000	0.000	0.001 [0.008]	0.001 [0.004]
	0.004	0.000	0.000	0.000 [0.001]	0.001 [0.001]
	0.050	0.041	0.044	0.041 [0.035]	0.064 [0.030]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.024	0.026	0.019	0.015 [0.057]	0.035 [0.062]
	0.016	0.018	0.015	0.010 [0.017]	0.035 [0.024]
	0.023	0.039	0.031	0.017 [0.057]	0.042 [0.062]
$\beta_0 - \beta_4$	0.452	0.634	0.605	0.605 [0.299]	0.866 [0.406]
	0.636	0.567	0.539	0.567 [0.197]	0.868 [0.296]
	0.769	0.738	0.737	0.753 [0.568]	0.839 [0.570]
$\beta_0 - \beta_5$	0.328	0.128	0.134	0.124 [0.148]	0.070 [0.170]
	0.112	0.103	0.108	0.111 [0.102]	0.065 [0.136]
	0.032	0.047	0.043	0.035 [0.058]	0.025 [0.068]
$\beta_0 - \beta_6$	0.076	0.093	0.096	0.092 [0.114]	0.017 [0.113]
	0.076	0.110	0.113	0.091 [0.129]	0.019 [0.154]
	0.042	0.039	0.041	0.046 [0.066]	0.015 [0.077]
$\beta_0 - \beta_7$	0.040	0.050	0.056	0.064 [0.107]	0.009 [0.086]
	0.044	0.080	0.080	0.082 [0.169]	0.010 [0.167]
	0.033	0.040	0.040	0.043 [0.070]	0.011 [0.080]
$\beta_0 - \beta_8$	0.024	0.031	0.041	0.049 [0.103]	0.002 [0.063]
	0.048	0.071	0.078	0.072 [0.192]	0.002 [0.142]
	0.025	0.026	0.028	0.032 [0.070]	0.001 [0.065]
$\beta_0 - \beta_9$	0.052	0.038	0.049	0.050 [0.193]	0.000 [0.055]
	0.064	0.051	0.067	0.067 [0.055]	0.000 [0.081]
	0.026	0.025	0.032	0.033 [0.067]	0.000 [0.043]

5.6 Variable Selection in the Presence of Outliers

The complexity of variable selection significantly increases for regression models contaminated with outliers. Markatou et al. (1991) state that tests on least squares regression parameters lose power dramatically in the presence of outliers and leverage points. One approach to overcome the loss of power is to use a robust regression estimator. The previous chapters illustrate that least squares is not the estimator of choice in contaminated samples and that compound estimators demonstrate the best overall performance. This section reviews the robust regression variable selection literature for both analytical and resampling methods and conducts comparative evaluations of resampling methods with compound estimators. There are few empirical results in the literature that address the combined problem of compound estimation and resampling because both procedures are computationally complex.

5.6.1 Variable Selection with Robust Regression Estimators

Although numerous robust estimators have been proposed in the last 25 years, there are significantly fewer results in the literature that explore variable selection procedures in the robust regression model. Most robust regression variable selection methods are based on robust versions of the general linear test that use the asymptotic covariance matrix (Hampel et. al, 1986). Markatou and He (1994) and Hertier and Ronchetti (1994) extend the Wald (similar to t-tests) and drop-in-dispersion tests (similar to F-tests) to GM and compound estimators. Field (1997) and Field and Welsh (1998) propose saddlepoint approximations of tail area probabilities for robust regression

hypothesis testing as improvements to the asymptotic approach. The results are mixed and they recommend further testing in finite samples. Ronchetti and Staudte (1994) propose a robust version of Mallows's C_p. The method multiplies the squared residuals by the final weights from a robust fit to compute the residual sum of squares. Two additional quantities are also added to the residual sum of squares that are a function of the number of parameters and the selected robust estimator. The robust C_p appears to work satisfactorily for their three examples, but no simulation results are reported.

The Wald test is currently preferred (Hertier, 1997) because of its asymptotic chi-square distribution and the relative ease to calculate the asymptotic covariance matrix. Wilcox (1997) experiments (results not reported) with the Wald test using the M-estimator and the Coakley and Hettmansperger (1993) compound estimator. For both estimators, he found poor control over the Type I error, even with normal error terms and $n = 100$. All authors conclude that it is important to do further testing and evaluation to understand the strengths and weaknesses of the methods in finite samples.

Bootstrap methods can be used in robust regression to construct confidence intervals and prediction intervals (Efron and Tibshirani, 1993, Davison and Hinkley, 1997, Wilcox, 1994, 1996a, 1996b, 1997). Mammen (1993) shows the consistency of the bootstrap for linear tests with the M estimator.

Wilcox (1997, 1998) presents an interesting approach to the variable selection problem in robust regression by using a bootstrap resampling scheme. He uses a percentile bootstrap approach to find critical values for the joint confidence region on the Mahalanobis distance for the model parameters. The steps of the algorithm are:

1. Obtain B bootstrap estimates of β by bootstrapping pairs.
2. Estimate the covariance matrix V using all B bootstrap estimates of β.
3. Find the Mahalanobis distance of $(\beta^* - \hat{\beta})$ using V^{-1} for each bootstrap sample where β^* is the bootstrap estimate of the model parameters and $\hat{\beta}$ is the vector of parameter estimates from the original data.
4. Sort the Mahalanobis distances and call the $(1-\alpha)B$ ordered distance the critical value.
5. Find the test statistic by the Mahalanobis distance using V^{-1} of $(\hat{\beta} - c)$ where c is a vector of constants often selected as 0 to test for significance.

Wilcox (1998) states that there is room for improvement with this method because the probability of a Type I error can be substantially less than nominal levels in many circumstances. He cautions that this approach does not work well with least squares; correction factors through simulation are required to achieve the correct coverage probabilities. Our experiments with compound estimators indicate that the algorithm is a dependable diagnostic to test if at least one of the variables is active; however, the test statistics are not useful to differentiate between competing models.

Davison and Hinkley (1997) provide a brief discussion of resampling methods in robust regression. Their guidance on resampling methods for variable selection in robust regression focuses on two main points: 1) remove gross outliers from the analysis because too many outliers could appear in the resampled data leading to inefficiency and breakdown and 2) most of the prediction error methods for least squares *should* apply to robust regression. Outliers are removed by residuals from an LTS fit.

Thus, there is relatively little guidance for variable selection using cross-validation or bootstrap estimates of prediction error in robust regression. The next section revisits and modifies the Gunst and Mason data to contain residual outliers. We

compare the same resampling methods and criteria as in Section 5.5, except that we use a compound estimator rather than least squares.

5.6.2 Modified Gunst and Mason Data

The Gunst and Mason data used in the previous sections have several observations that are extreme in X-space. The hat diagonals indicate that only 4 of the 40 observations (2, 8, 15, and 39) are remote in X-space using the usual $3p/n$ criteria (Hoaglin and Welsch, 1978). However, the Rocke and Woodruff (1996) robust distances (see Chapter 4) in Table 5.8 conclude that the 12 shaded observations are remote in X-space. These high-leverage points could also explain the large R^2 values seen in Section 5.4. In practice, the response values of these extreme points in X-space may not follow the regression surface as well as in the previous experiments. We plant four residual outliers by adding 10.0 to the response values of the high-leverage observations 8, 15, 28, and 39. The data set now contains 10% residual outliers at a distance of 10σ. The simulations are run exactly as described in Section 5.4 and $\beta' = [2, 0, 0, 4, 8]$.

Table 5.8. Rocke and Woodruff (1996) robust distances for the Gunst and Mason (1980) data. The observations with shaded robust distance cells are considered remote in X-space because they exceed the cutoff value of 10.

obs	RD	obs	RD	obs	RD	obs	RD	obs	RD	obs	RD	obs	RD	obs	RD
1	11	6	3	11	6	16	2	21	5	26	8	31	24	36	7
2	777	7	77	12	7	17	6	22	1	27	2	32	3	37	1
3	1	8	426	13	4	18	1	23	1	28	191	33	41	38	3
4	28	9	9	14	1	19	6	24	3	29	3	34	101	39	79
5	2	10	2	15	132	20	2	25	8	30	3	35	115	40	1

The probabilities in Table 5.9 indicate that no resampling technique using any criterion successfully selects the correct three-parameter model. The two inactive variables are now significant in model selection because the least squares estimator has used them to fit the outliers. This example illustrates the important linkage between outlier identification and variable selection in model building. The planted outliers are masked; they do not have unusual residual values. Note, also from Table 5.9, that the minimum prediction error criterion most often selects the 5 parameter model while the change in prediction error criterion selects the 4 parameter model. The least squares estimator has failed; outliers are masked and insignificant variables now appear to be significant.

A logical choice for this data set contaminated with high-leverage points and residual outliers is a compound estimator. Resampling estimates of prediction error with compound estimators could potentially pose some problems. For example, the estimator could breakdown because, by chance, a bootstrap sample may contain too many of the planted outliers. Breakdown means that the parameter estimates are no longer valid for the bulk of the data (see Chapter 2). Also, if the compound estimator successfully downweights the outliers, the resulting prediction error may not necessarily be low relative to the other models. This is explained by the existence of two sources of error contributing to overall prediction error. One source of error is the lack of a good fit due to model misspecification. The other source is that the estimator works and assigns large residual values to the outliers which inflates the overall prediction error.

Table 5.9. Results for 1000 simulations for model selection from Shao (1996) using least squares parameter estimates. Observations 8, 15, 28, and 39 are made residual outliers. The true model [2, 0, 0, 4, 8] is shaded. The top number in each cell is the fraction of time the model was selected using the minimization of prediction error metric and the bottom uses the change in purity metric with constant 0.025. The values in brackets for the bootstrap are the ratios out of 100,000 bootstrap samples that the model is selected.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.001]	0.000 [0.013]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.000]
β_0, β_3	0.000	0.000	0.000	0.000 [0.020]	0.000 [0.000]
	0.000	0.000	0.000	0.000 [0.072]	0.000 [0.000]
$\beta_0, \beta_3, \beta_4$	0.000	0.000	0.000	0.000 [0.037]	0.000 [0.110]
	0.024	0.000	0.000	0.335 [0.302]	0.269 [0.518]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.797	0.055	0.037	0.116 [0.275]	0.215 [0.304]
	0.976	1.000	1.000	0.665 [0.398]	0.727 [0.303]
$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	0.203	0.945	0.963	0.884 [0.667]	0.785 [0.572]
	0.000	0.000	0.000	0.000 [0.227]	0.004 [0.179]

The values in Table 5.10 are the proportion of 100 replicates that the model was selected if the Simpson and Montgomery (1998a) compound estimator replaces least squares. The change in prediction error criterion (constant = 0.025) reliably identifies the correct model for all resampling methods with a slight edge given to the bootstrap full sample. The minimum prediction error criterion is not useful except for the bootstrap. Note that the minimum prediction error criterion performs poorly with the full sample bootstrap.

Table 5.10. Results for 100 simulations for model selection from Shao (1996) using the Simpson and Montgomery compound estimator. Observations 8, 15, 28, and 39 are made residual outliers and the estimator is Simpson and Montgomery. The true model [2, 0, 0, 4, 8] is shaded. The top number in each cell is the fraction of time the model was selected using the minimization of prediction error metric and the bottom number uses the change in purity metric with constant 0.025. The values in brackets for the bootstrap are the ratios out of 100,000 bootstrap samples that the model is selected.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross–Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.001]	0.000 [0.023]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.001]
β_0, β_3	0.000	0.000	0.000	0.000 [0.113]	0.000 [0.007]
	0.000	0.000	0.000	0.000 [0.163]	0.000 [0.006]
$\beta_0, \beta_3, \beta_4$	0.450	0.490	0.480	0.000 [0.067]	0.930 [0.429]
	0.980	0.940	0.930	0.990 [0.449]	0.980 [0.685]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.290	0.290	0.250	0.410 [0.242]	0.070 [0.283]
	0.000	0.020	0.010	0.010 [0.237]	0.010 [0.120]
$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	0.260	0.260	0.270	0.590 [0.578]	0.000 [0.258]
	0.020	0.040	0.060	0.000 [0.151]	0.010 [0.189]

As an alternative to resampling with a computationally inefficient compound estimator, we consider the Davison and Hinkley (1997) recommendation to first remove large residual observations from a robust fit. Their choice of estimators, LTS, is high-breakdown but is not a bounded-influence estimator. Therefore, outliers will likely be removed from the sample only if they are not high-leverage points.

For the modified Gunst and Mason data, we first remove the observations with standardized residuals exceeding a value of 2.5 from a fit with the high-breakdown, high-efficiency, and bounded-influence Simpson and Montgomery compound estimator. Subsequently, resampling methods estimate the least squares prediction error for variable selection. The Simpson and Montgomery filter removes the outlying observations at the beginning of every one of the 100 replications. From Table 5.11, this scheme

successfully identifies the correct model with high probability for the proposed change in prediction error criterion. The minimum prediction error criterion has a high probability of correct model selection only for the bootstrap half sample.

Table 5.11. Results for 100 simulations for model selection from Shao (1996) using the Simpson and Montgomery compound estimator to remove outliers followed by least squares estimates. Observations 8, 15, 28, and 39 are made residual outliers and the estimator is Simpson and Montgomery. The true model [2, 0, 0, 4, 8] is shaded. The top number in each cell is the proportion of 100 replicates that the model was selected using the minimization of prediction error criterion and the bottom number is the proportion with the change in prediction error criterion with constant 0.025. The values in brackets are the proportion of 10,000 bootstrap samples that the model is selected.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.014]
	0.000	0.000	0.000	0.000 [0.000]	0.000 [0.001]
β_0, β_3	0.000	0.000	0.000	0.000 [0.002]	0.000 [0.002]
	0.000	0.000	0.000	0.000 [0.087]	0.000 [0.006]
$\beta_0, \beta_3, \beta_4$	0.670	0.580	0.580	0.720 [0.447]	0.910 [0.535]
	0.950	0.950	0.950	0.990 [0.868]	0.960 [0.685]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.110	0.170	0.180	0.110 [0.213]	0.040 [0.225]
	0.010	0.010	0.010	0.010 [0.013]	0.010 [0.120]
$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	0.220	0.050	0.240	0.170 [0.337]	0.050 [0.224]
	0.040	0.040	0.040	0.000 [0.033]	0.030 [0.189]

The fit for the clean 36 observations has an R^2 of approximately 0.99, much like the scenarios of Section 5.4. If the random error added to the response variable is generated from variates of NID(0, 5^2) instead of NID(0, 1), then the Simpson and Montgomery compound estimator (and all other robust regression estimators) fails to provide meaningful parameter estimates. The parameter estimates vary widely between the simulation replicates and significantly within the bootstrap samples. None of the resampling methods or criteria reliably identify the specified model.

5.6.3 Compound Estimator Resampling Methods for a Noisy, High-Dimension Data Set with Multiple Outliers

To investigate the performance of the resampling methods in the presence of outliers and noisy data, we generate an artificial data set. The response is generated from $y = X\beta + \varepsilon$ where X is a 40 x 9 matrix of standard normal variates augmented with a column of ones, the known vector of parameters β' is [2, 9, 6, 4, 8, 0, 0, 0, 0, 0] and ε is NID($0, \sigma^2 I$) with $\sigma = 5$. The last five observations are 10σ residual outliers and the last three observations are also 10σ outliers in X-space for variables x_3 through x_6.

In contrast to all previous findings for the minimum prediction error criterion, the bootstrap half sample results in Table 5.12 do not improve upon those from the full sample. The change in prediction error criterion is successful for all methods except the bootstrap half sample. The minimum prediction error criterion does not perform well with cross-validation.

Table 5.12. Results for 50 simulation replicates for model selection with the Simpson and Montgomery estimator. All regressor variable values are generated from a standard normal distribution. The response is generated from the vector [2, 9, 6, 4, 8, 0, 0, 0, 0, 0] and ε NID $(0, \sigma^2 I)$ with $\sigma = 5$. The last five observations are 10σ residual outliers and the last three observations are also 10σ outliers in X-space for variables x_3 through x_6. The top number in each cell is the proportion of 50 replicates that the model was selected using the minimization of prediction error criterion and the bottom number is the change in prediction error criterion with constant 0.01. The values in brackets are the ratios out of 5,000 bootstrap samples that the model is selected. Results are accurate to approximately ± 0.06.

Model parameters	Cross-Val Lv 1 out	Cross-Val K = 6	Adj Cross – Val K=6	Bootstrap Full Sample (n=40)	Bootstrap Half Sample (n=20)
β_0	0.000	0.000	0.000	0.000 [0.004]	0.000 [0.037]
	0.000	0.000	0.000	0.000 [0.001]	0.000 [0.000]
β_0, β_3	0.000	0.000	0.000	0.000 [0.016]	0.000 [0.000]
	0.000	0.000	0.000	0.000 [0.008]	0.000 [0.004]
$\beta_0, \beta_3, \beta_4$	0.000	0.000	0.000	0.000 [0.022]	0.000 [0.000]
	0.000	0.000	0.020	0.000 [0.009]	0.000 [0.003]
$\beta_0, \beta_1, \beta_3, \beta_4$	0.000	0.000	0.000	0.000 [0.035]	0.060 [0.029]
	0.000	0.000	0.000	0.000 [0.031]	0.000 [0.016]
β_0 - β_4	0.540	0.360	0.420	0.780 [0.340]	0.720 [0.418]
	0.820	0.820	0.840	0.900 [0.549]	0.560 [0.382]
β_0 - β_5	0.180	0.380	0.300	0.180 [0.215]	0.220 [0.219]
	0.060	0.020	0.020	0.040 [0.061]	0.100 [0.068]
β_0 - β_6	0.040	0.100	0.100	0.020 [0.138]	0.000 [0.111]
	0.020	0.060	0.040	0.000 [0.073]	0.060 [0.112]
β_0 - β_7	0.100	0.060	0.040	0.020 [0.091]	0.000 [0.082]
	0.040	0.040	0.020	0.060 [0.101]	0.220 [0.187]
β_0 - β_8	0.060	0.000	0.000	0.000 [0.078]	0.000 [0.056]
	0.000	0.020	0.020	0.000 [0.091]	0.040 [0.131]
β_0 - β_9	0.080	0.100	0.140	0.000 [0.062]	0.000 [0.048]
	0.060	0.040	0.040	0.000 [0.076]	0.020 [0.098]

5.6.4 A Designed Experiment for Resampling Methods with Compound Estimators

Based on the modified Gunst and Mason data and the artificial data set in Section 5.6.3, it appears as if resampling methods are appropriate for variable selection when outliers are present. To gain a better understanding of resampling methods' performance

for the variable selection problem with multiple outliers, we run a designed experiment using Monte Carlo simulation. The experiment varies characteristics of not only the data set, but also of the resampling method to quantify the expected performance of the various techniques. We use the Simpson and Montgomery compound estimator for all simulations. Note that we are not removing the outliers from analysis first as recommended by Davison and Hinkley (1997) and explored in Table 5.11.

5.6.4.1 Planning the Simulation Experiment

All data sets consist of $n = 40$ observations and $p = 5$ parameters. The response vector is generated as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where $\mathbf{X}$ is the design matrix of i.i.d. random variates from the standard normal distribution, $\boldsymbol{\beta}'$ is the vector of known parameters [2, 4, 8, 0, 0], and $\boldsymbol{\varepsilon}$ is the vector of random error variates from a $N(0, \sigma_e^2\mathbf{I})$ distribution. For the last four observations, a value of δ is added to each regressor variable value to create high-leverage points. Residual outliers are created for the last four or eight observations (depending on the factor setting) by adding δ to the expected response value.

Factors for the Experiment. From the previous results, pilot studies, and knowledge of compound estimators, the following factors are included:

- Percentage of outliers contaminating the sample. This could be an important factor because resampling methods could form samples with too many outliers that cause the estimator to break down. Also, prediction error is higher with more outliers. The outlier density levels are 10% and 20%.

- Outlying distance, δ. This measures how many standard deviations to make the outliers; both in leverage and residual. Larger values could lead to greater prediction error. The levels are 5 standard deviations and 10 standard deviations.

- Signal-to-noise ratio, measured through σ_e. Section 5.5 demonstrates that this is a critical factor in determining the success of a procedure. The probability of correctly selecting the model is directly proportional to the signal-to-noise ratio. The levels for σ_e are 1 and 5 which corresponds to approximate R^2 values of 0.98 and 0.80 respectively for an OLS fit on the uncontaminated portion of the data.

- Bootstrap sample size. Shao (1996) demonstrates that this is the single most important factor to correctly identify the active model parameters. Sections 5.4 and 5.5 also indicate the half sample size is preferred. However, there may not be an appreciable difference for contaminated data sets (see Table 5.12). The levels again are the full sample ($n = 40$) and the half sample ($n = 20$).

- Number of bootstraps per replication. Up to this point, we have followed Davison and Hinkley's (1997) recommendation to use 100 bootstrap samples as an absolute minimum. Breiman and Spector (1992) and Breiman (1996) do not exceed 50 bootstrap samples and conclude for some applications that as few as 5 may suffice. Clearly, fewer bootstraps than 100 would be preferred

when resampling with a compound estimator. The levels in the experiment are B = 25 and 100 bootstrap samples.

- Size of assessment set in cross-validation. This factor replaces the previous two bootstrap-specific factors in the cross-validation runs. The purpose of this factor is to determine if there is a significant difference between K-Fold and leave-one-out cross-validation procedures. The levels are 6 (K-Fold) and 1 (leave-one-out).

Experimental Design and Response. All five factors for the bootstrap have only two levels; therefore, an attractive screening design for this experiment is a 2_V^{5-1} design. This design can estimate the main effects and the two-factor interactions free from aliasing. The cross-validation design is a full factorial 2^4. The response value again is the proportion of replicates in which the various parameter models are selected. We also investigate the usefulness of weighting the squared residuals by the final weights from the Simpson and Montgomery estimator. The motivation for this additional response comes from Ronchetti and Staudte's (1994) robust C_p criterion and from pilot experiments that showed a significant amount of the prediction error could be attributed to the large residuals of the planted outliers.

Pilot studies are necessary to select an appropriate value for the constant for the change in prediction error criterion. The three factors that affect the choice of the constant are the outlying distance, signal-to-noise ratio and weighting of the residuals. Table 5.13 gives the values for the constants used in the simulations.

Table 5.13. Values of constants used in simulations
for the change in prediction error criterion.

δ	σ_e	Constant for unwtd	Constant for weighted
5	1	0.0250	0.0250
10	1	0.0250	0.0100
5	5	0.0050	0.0010
10	5	0.0025	0.0005

5.6.4.2 Simulation Results

Bootstrap Methods.   The most striking aspect of the probabilities in Table 5.14 is the contrast between the left half and the right half of the table. This corresponds to the difference between prediction errors computed with the unweighted versus weighted residuals. Weighting the residuals leads to nearly perfect selection of the correct model using the change in prediction error criterion independent of any other factor setting. Conversely, this weighting scheme almost always incorrectly selects the largest model if the minimum prediction error criterion is used. This suggests some modifications could be made to the prediction error calculation under a weighting scheme if the minimum prediction error criterion is used. The modified calculation could account for the number of model parameters; similar to the robust C_p.

If the residuals are not weighted (the left half of Table 5.14), the most important factor under either selection criterion is the amount of noise used to generate the response values. The correct model is selected with virtual certainty if the signal-to-noise ratio is high ($\sigma_e = 1$) under the change in prediction error criterion. In fact, this is the only

significant factor from the ANOVA ($R^2 = 0.85$) for unweighted residuals with the change in prediction error criterion. For the unweighted residuals with minimum prediction error criterion, the significant effects from ANOVA ($R^2 = 0.87$) are the signal-to-noise, the bootstrap sample size, and the number of bootstrap samples. Better model selection occurs with smaller bootstrap sample sizes, larger bootstrap samples and, surprisingly, lower signal-to-noise ($\sigma_e = 5$). The change in prediction error criterion significantly outperforms the minimum prediction error criterion with unweighted residuals in high signal-to-noise scenarios and is moderately outperformed in the low signal-to-noise scenarios.

Table 5.14. 2_V^{5-1} design and results for bootstrap methods using compound estimators. The top values in each cell are the proportion of times that the model is selected out of 50 replications using the minimum prediction error criterion. The bottom values are the change in prediction error criteria.

% out	δ	σ_e	Boot Size $n\text{-}m$	Num Boot	Boot $\beta_0\text{-}\beta_1$	Boot $\beta_0\text{-}\beta_2$	Boot $\beta_0\text{-}\beta_3$	Boot $\beta_0\text{-}\beta_4$	Boot wt $\beta_0\text{-}\beta_1$	Boot Wt $\beta_0\text{-}\beta_2$	Boot wt $\beta_0\text{-}\beta_3$	Boot wt $\beta_0\text{-}\beta_4$
10	5	1	20	100	0.000 0.000	0.740 0.980	0.260 0.000	0.000 0.020	0.000 0.000	0.000 1.000	0.000 0.000	1.000 0.000
20	5	1	20	25	0.000 0.000	0.680 0.960	0.260 0.020	0.060 0.020	0.000 0.000	0.000 1.000	0.020 0.000	0.980 0.000
10	10	1	20	25	0.000 0.000	0.540 0.960	0.400 0.040	0.060 0.000	0.000 0.000	0.000 1.000	0.000 0.000	0.000 0.000
20	10	1	20	100	0.000 0.000	0.740 0.940	0.240 0.000	0.020 0.060	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000
10	5	5	20	25	0.000 0.000	0.760 0.600	0.180 0.120	0.060 0.280	0.000 0.000	0.000 1.000	0.000 0.000	1.000 0.000
20	5	5	20	100	0.000 0.000	0.880 0.780	0.080 0.100	0.040 0.120	0.000 0.000	0.000 0.940	0.000 0.040	1.000 0.020
10	10	5	20	100	0.000 0.000	0.820 0.720	0.140 0.120	0.040 0.160	0.000 0.000	0.000 0.100	0.000 0.000	1.000 0.000
20	10	5	20	25	0.000 0.000	0.740 0.620	0.220 0.160	0.040 0.220	0.000 0.000	0.180 0.960	0.100 0.000	0.720 0.040
10	5	1	40	25	0.000 0.000	0.440 1.000	0.340 0.000	0.220 0.000	0.000 0.000	0.000 1.000	0.020 0.000	0.980 0.000
20	5	1	40	100	0.000 0.000	0.620 1.000	0.320 0.000	0.060 0.000	0.000 0.000	0.000 1.000	0.000 0.000	1.000 0.000
10	10	1	40	100	0.000 0.000	0.460 1.000	0.380 0.000	0.160 0.000	0.000 0.000	0.000 1.000	0.000 0.000	1.000 0.000
20	10	1	40	25	0.000 0.000	0.320 1.000	0.320 0.000	0.360 0.000	0.000 0.000	0.020 1.000	0.180 0.000	0.800 0.000
10	5	5	40	100	0.000 0.000	0.680 0.720	0.280 0.200	0.040 0.080	0.000 0.000	0.000 1.000	0.000 0.000	1.000 0.000
20	5	5	40	25	0.000 0.000	0.620 0.680	0.300 0.220	0.080 0.100	0.000 0.000	0.040 0.920	0.100 0.040	0.860 0.040
10	10	5	40	25	0.000 0.000	0.620 0.440	0.320 0.300	0.060 0.260	0.000 0.000	0.000 1.000	0.020 0.000	0.980 0.000
20	10	5	40	100	0.000 0.000	0.700 0.660	0.280 0.240	0.020 0.100	0.000 0.000	0.020 1.000	0.040 0.000	0.940 0.000

Cross-Validation Methods. The difference between using weighted and unweighted residuals is not as distinct with cross-validation methods as it is for the bootstrap. The minimum prediction error criterion using weighted residuals no longer selects exclusively the largest parameter model. It selects the correct model, independent of factor settings, between 40 and 50% of the time. The change in prediction error criterion with weighted residuals selects the correct model with very high probability if the signal-to-noise ratio is high; otherwise, it has about a 70% correct selection rate in lower signal-to-noise scenarios. Signal-to-noise ratio is the only significant variable from ANOVA ($R^2 = 0.85$) for the change in prediction criterion using weighted residuals.

If the residuals are not weighted, then the minimum prediction error criterion is still not effective; correct model selection probabilities are between 0.2 and 0.5. All four factors are significant for this criterion from ANOVA ($R^2 = 0.90$). Signal-to-noise ratio has the largest effect. The outlier magnitude and signal-to-noise ratio and their two-factor interaction are the significant factors ($R^2 = 0.95$) for the change in prediction error criterion with unweighted residuals. Performance of this criterion is similar to the weighted residuals case: near perfect model selection if the signal-to-noise ratio is high and about 70% otherwise (although considerably more variance between factor settings).

Clearly, the best method across all scenarios is the change in prediction error criterion applied to weighted prediction error from the bootstrap procedure. If the change in prediction error criterion is used with unweighted residuals, then cross-validation gives slightly better results than bootstrap procedures. For the minimum prediction error criterion, cross-validation is not recommended. The best results are from the bootstrap

half sample. There does not seem to be much difference between K-Fold and leave-one-out cross-validation procedures for either criterion.

Table 5.15. 2^4 design and results for cross-validation methods using compound estimators. The top values in each cell are the proportion of times that the model is selected out of 50 replications using the minimum prediction error criterion. The bottom values are the change in prediction error criteria.

% out	δ	σ_e	Size of $S_{k,a}$	CV β_0-β_1	CV β_0-β_2	CV β_0-β_3	CV β_0-β_4	CV wt β_0-β_1	CV wt β_0-β_2	CV wt β_0-β_3	CV Wt β_0-β_4
10	5	1	1	0.000	0.280	0.360	0.360	0.000	0.440	0.320	0.240
				0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
20	5	1	1	0.000	0.320	0.320	0.360	0.000	0.420	0.200	0.380
				0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
10	10	1	1	0.000	0.200	0.440	0.360	0.000	0.440	0.320	0.240
				0.000	1.000	0.000	0.000	0.220	0.780	0.000	0.000
20	10	1	1	0.000	0.280	0.300	0.420	0.000	0.440	0.240	0.320
				0.000	1.000	0.000	0.000	0.100	0.900	1.000	0.000
10	5	5	1	0.000	0.400	0.360	0.240	0.020	0.420	0.340	0.220
				0.000	0.800	0.100	0.100	0.020	0.760	0.160	0.060
20	5	5	1	0.000	0.520	0.200	0.280	0.000	0.460	0.240	0.300
				0.000	0.860	0.040	0.100	0.000	0.760	0.100	0.140
10	10	5	1	0.000	0.340	0.320	0.240	0.020	0.480	0.200	0.300
				0.020	0.540	0.240	0.200	0.000	0.640	0.200	0.160
20	10	5	1	0.000	0.480	0.240	0.280	0.040	0.460	0.240	0.260
				0.040	0.640	0.160	0.160	0.000	0.700	0.100	0.120
10	5	1	6	0.000	0.280	0.320	0.400	0.000	0.480	0.280	0.240
				0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
20	5	1	6	0.000	0.420	0.220	0.360	0.000	0.480	0.140	0.380
				0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
10	10	1	6	0.000	0.260	0.320	0.420	0.000	0.380	0.420	0.200
				0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
20	10	1	6	0.000	0.320	0.260	0.420	0.000	0.420	0.220	0.360
				0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
10	5	5	6	0.000	0.520	0.240	0.240	0.060	0.420	0.260	0.260
				0.000	0.700	0.180	0.120	0.040	0.680	0.140	0.140
20	5	5	6	0.000	0.540	0.180	0.280	0.020	0.460	0.220	0.300
				0.000	0.780	0.120	0.100	0.020	0.700	0.100	0.180
10	10	5	6	0.000	0.500	0.280	0.220	0.040	0.480	0.260	0.220
				0.000	0.620	0.220	0.160	0.000	0.700	0.180	0.120
20	10	5	6	0.000	0.440	0.300	0.260	0.000	0.460	0.220	0.320
				0.020	0.600	0.240	0.140	0.000	0.680	0.160	0.160

5.7 Summary

This chapter proposes criterion for model selection as an alternative to the strict minimization of prediction error. A criterion that selects the model that has the fewest variables and low prediction error is often a better choice. To implement and test this procedure, an operational version is introduced that increases the dimension of the model until the change in prediction error is less than a specified percentage of total prediction error in the intercept only model. Extensive Monte Carlo simulation suggests this criterion often outperforms the minimum prediction error criterion in both contaminated and uncontaminated samples. The criterion is tested using prediction error estimates from the leave-one-out cross-validation, K-Fold cross-validation, adjusted K-Fold cross-validation, the bias adjusted bootstrap, and the bootstrap half sample procedures.

5.7.1 Summary of Results for Least Squares Estimation

- For the Shao (1996) scenarios, the proposed criterion has nearly a 100% correct model selection rate for all resampling procedures because the signal-to-noise is very high ($R^2 = 0.99$) and there are only 4 regressor variables. Only the bootstrap half sample method is consistently above 80% using the minimum prediction error criterion.

- If the Shao (1996) data set is extended to 9 regressor variables, then the proposed criterion exceeds a 91% correct selection rate for all five resampling methods. The minimum change in prediction error selection rate is below 70% for all procedures except the bootstrap half sample (89%).

- If the signal-to-noise ratio is decreased ($R^2 = 0.70$) in the 9 regressor variable model, then the proposed criterion correct selection rate is approximately 75% for all methods except the bootstrap half sample (84%). The minimum prediction error criterion selection rate is below 63% for all methods except the bootstrap half sample (87%). This shows that methods other than the bootstrap half sample are competitive when the proposed criterion is used.

5.7.2 Summary of Results for Compound Estimation

- If 10% residual outliers are planted in the Gunst and Mason data set (already contaminated with high-leverage values), then all methods and criteria fail to identify the correct model with least squares. If the least squares estimator is replaced with the Simpson & Montgomery compound estimator, then the proposed criterion selects the correct model over 93% of the time for all resampling methods. The minimum prediction error criterion has below a 50% correct selection rate except for the bootstrap half sample procedure (93%).

- If the number of regressors increases to 9 in the modified Gunst and Mason data and the signal-to-noise ratio decreases ($R^2 = 0.80$), then the proposed criterion selects the correct model over 80% of the time for cross-validation and 90% for the bootstrap. The minimum change in prediction error is below 80% for the bootstrap and below 55% for cross-validation. Most importantly, the bootstrap half sample is worse than the full sample.

- A designed experiment investigating the effect of outlier density, outlier magnitude, signal-to-noise ratio, and sample sizes for the resampling methods demonstrates that the proposed criterion is preferable or comparable to the minimum prediction error criterion. If the residuals are weighted by the final weights from the compound estimator, the correct model is almost always selected with the proposed criterion for the bootstrap methods. However, in the same scenarios, the minimum prediction error criterion always overfits using weighted residuals. For unweighted residuals, the proposed criterion is often preferred and always competitive for all scenarios.

Chapter 6

Summary, Contributions, and Future Research

6.1 Introduction

This research uses extensive Monte Carlo simulation to evaluate several aspects of the multiple outlier problem in regression. Chapter 1 demonstrates the impact that multiple outliers can have on a regression model, the failure of standard OLS diagnostic measures to detect the outliers, and the trouble outliers can cause to the variable selection process. The stated objectives of this research are to comprehensively test the leading multiple outlier detection procedures, improve existing methods that identify and accommodate outliers and investigate the usefulness of resampling methods for variable selection in regression models with multiple outliers. These three objectives are addressed in Chapters 3-5 respectively. This chapter provides a summary of the major findings for each objective, the original contribution, and recommendations for future research.

6.2 Comparative Analysis of Multiple Outlier Detection Procedures

The objective is to conduct a comprehensive performance study of numerous multiple outlier detection methods proposed in the literature. The methods are tested in realistic and challenging regression scenarios to establish the candidates' strengths and weaknesses.

6.2.1 Summary of Significant Findings

The single most important factor affecting the performance of all methods is the leverage of the outlying observations. The significant results are reported for high-leverage (exterior X-space) and low-leverage (interior X-space) outliers. Many procedures have not previously been tested with high-leverage outliers.

Low-leverage outliers. All of the selected methods (except Pena and Yohai) perform well for low-leverage outliers once the outlying distance exceeds 5σ of the regression surface. OLS generally detects the outliers, but suffers from significant false alarms as the magnitude of the outlying distance increases. The indirect procedures dominate the direct methods with one notable exception. The Sebert et al. clustering methodology is in many cases the best method; however, the false alarm rate can be high and some scenarios defeat the method. Overall, the high-breakdown point (HBP) estimators are recommended; in particular, the *MM* estimator. For all procedures, the factor with the greatest impact, apart from leverage, is outlying distance followed by outlier density and dimension, respectively.

High-leverage outliers. The HBP estimators that are successful in the low leverage scenarios perform poorly if the outliers are also remote in X-space. Most direct procedures lose a significant amount of detection capability with the high-leverage points because the algorithms rely on a least squares residuals. The compound robust regression estimators are generally preferred to the direct algorithms. The Simpson & Montgomery compound estimator has the best overall performance. Also, the Rousseeuw and van Zomeren method using simulated cutoff values is powerful. This suggests that the newer

MVE and LMS algorithms are not plagued as much by the criticisms of the random sampling schemes. For all methods, the most significant factors affecting performance are the leverage and the residual magnitude and their two-factor interaction.

6.2.2 Contributions

Several multiple outlier detection procedures have been proposed in recent years. All demonstrate good results in the authors' limited studies that are often restricted to "classic data sets" or low-dimension, low-leverage examples. There has not been a comprehensive evaluation of procedures since 1990. Every method tested in this research has been proposed since 1990. The contributions are:

- A direct comparison of the current multiple outlier detection methods.

- Sensitivity analysis of all procedures to outlier magnitude, density, leverage, and configuration in X-space.

- The recommendation that robust regression estimators are in most cases superior to the direct methods. It may be of little use to integrate one of the specialized direct methods into a suite of regression analysis tools. Robust regression capability is all that is required.

6.2.3 Future Research

Monte Carlo simulation is the method used to evaluate performance in the selected outlier scenarios. These scenarios are limited to mean shift outliers and typically multiple point clouds. Performance studies with other approaches to data generation

would be useful to further test the procedures. The results from this research could be used to screen the multiple outlier detection methods and alternative outlier scenarios could be run (e.g. Breiman and Spector, 1992, Rocke and Woodruff, 1996, Wilcox, 1996a).

The research in Chapter 3 shows that in general the robust regression estimators outperform the direct methods. As such, Chapter 4 explored ways to improve the compound estimators. It is likely that some of the direct methods could be improved by integrating a robust estimator into the process. For example, most direct methods suffer significant loss in power for the high-leverage scenarios. This often can be traced back to the method depending on some form of the least squares residual driving the algorithm.

Two recent multiple outlier detection methods (Lee and Fung, 1997, and Luceno, 1998) address the generalized linear model (GLIM). There are no results in the literature that compare detection methods for the GLIM. Furthermore, many of the concepts for the direct identification methods and robust estimators from this research could be applied to the GLIM. Improved methods could be proposed for the GLIM.

6.3 An Improved Compound Estimator

The second research objective is to use the results from the performance study in Chapter 3 and improve upon an existing technique. The mechanics of compound estimators are evaluated more closely because of their favorable performance with high-leverage outliers in the comparative analysis. The leading compound estimators have vulnerability in high-dimension, high-leverage and high-density scenarios. Two

characteristics of the compound estimators are noted in these scenarios that require closer scrutiny. The π-weights are not unusual for the high-leverage outliers and the final parameter estimates do not differ significantly from the initial estimates.

6.3.1 Summary of Significant Findings

Performance study on measures of leverage. This Monte Carlo simulation study compares the Mahalanobis distance (hat diagonal), MVE, MCD, Hadi sequential point addition algorithm, Sebert et al. clustering methodology, *M*-estimates of covariance, and Rocke and Woodruff hybrid algorithm to identify remote observations in X-space. Mahalanobis distance breaks down in nearly all tested scenarios and the *M*-estimates of Covariance performs only slightly better. The Hadi algorithm can be tuned for excellent performance except in high-dimension. The MVE and MCD have comparable performance to one another; the MCD demonstrates slightly better results overall. The Sebert et al. method performs well, but can be vulnerable when the predicted response values are not Y-space outliers. Overall, the Rocke and Woodruff method demonstrates the best results for detection capability and resistance to false alarms.

A new measure of leverage in published compound estimators. Incorporation of the Rocke and Woodruff robust distances in the Coakley and Hettmansperger and Simpson and Montgomery compound estimators does not improve the performance in the vulnerable scenarios. The final weights are slightly unusual for the outliers if the new leverage measure is used. However, if the number of iterations of IRLS is increased to 3

or 4, then the outliers are properly assigned large residual values and the regression surface is not pulled toward the outliers.

Initial estimator study. In many high-leverage scenarios, the high-breakdown estimator provides poor estimates in the first stage of a compound estimator. High-breakdown point estimators do not have bounded-influence. An initial estimator is proposed that removes only the high-leverage and the high-residual observations from the sample, rather than 50% of the observations as the common high-breakdown estimators often do. High-leverage points are removed from the sample if the Rocke and Woodruff robust distance values exceed the cutoff value. Next, the residual outliers are removed if the standardized residual from an *MM* fit exceeds approximately 2.0. Lastly, an OLS fit on the remaining observations provides the parameter estimates. This is an efficient, high-breakdown, and bounded-influence initial estimator. Testing indicates that this estimator is highly successful not only in the high-density, high-dimension and high-leverage scenarios, but also all other outlier configurations.

Proposed compound estimator. The proposed compound estimator uses the new initial estimator and also the improved Rocke and Woodruff robust distances for the π-weight component. It significantly expands the effective region of operability for compound estimation with respect to outlying distance in both leverage and residual. Also, the estimator performs well in a published comparative analysis of robust regression estimators (Simpson and Montgomery, 1998b) where the leverage distances are not as challenging.

6.3.2 Contributions

- A comprehensive performance study for measures of leverage. Published results are limited to certain estimators and specific scenarios. There are no results on the performance of the MVE and MCD with the increased efficiency algorithms; Simpson and Chang (1997) call for such a study.

- An efficient, bounded-influence, and high-breakdown initial estimator. All initial estimators are high-breakdown only and may not provide useful parameter estimates in high-leverage scenarios. A good initial estimate is essential to a compound estimator because the final parameters may not change much and the final scale estimate is often based on the initial estimator's residuals.

- An improved compound estimator. The proposed estimator expands the area of coverage in high-dimension. Hampel (1997) states that a major gap in robust statistics is the lack of results and available tools for high-dimension.

6.3.3 Future Research

The proposed initial and compound estimator could be used as indirect methods for multiple outlier detection. Pilot studies show that these methods detect the planted outliers in the scenarios of Chapter 3 where all other methods fail. Additional finite sample performance studies are needed; especially in high-dimension. An improved plot

to the Rousseeuw and van Zomeren robust distances from the MVE and standardized LMS residuals is possible by replacing these measures with the Rocke and Woodruff robust distances and the proposed compound estimator's standardized residuals. Possibly some clustering of these components akin to Sebert et al. could be useful for outlier identification.

A more critical evaluation of the components of the compound estimators could be beneficial. Specifically, some studies can be done on the best way to form the π-weights from the Rocke and Woodruff robust distances. Also, this research did not consider the impact of changing the ψ function and estimates of scale. Another opportunity for improvement is to follow the Simpson and Chang (1997) recommendation to use a Hill-Ryan GM objective function rather than Schweppe or Mallows.

6.4 Resampling Methods for Variable Selection

The last research objective is to determine the appropriateness of resampling methods for variable selection in the presence of multiple outliers. Resampling methods with cross-validation and bootstrap estimates of model prediction error are currently the preferred approach to variable selection in OLS. Their major drawback is that they are computationally intense. Robust regression estimators are also computationally intense. With computational power increasing at dramatic rates, it will not be long before using resampling methods with robust regression is a viable approach for the practitioner. This research explored combining these two classes of procedures.

6.4.1 Summary of Significant Findings

An alternative variable selection criterion for OLS. All of the proposed regression variable selection procedures with resampling methods suggest that the best model is the one with the minimum prediction error. This research proposes a more realistic criterion that selects the model with the fewest parameters and a low (not necessarily minimum) prediction error. The scenarios in Shao (1996) are rerun using the proposed criterion. The results indicate that the proposed criterion is superior to the minimum prediction error criterion. Therefore, a bootstrap procedure using bootstrap sample sizes of less than ½ the original sample is not the only method to select the appropriate size model. The proposed procedure also works well for cross-validation procedures and the bootstrap using the full sample. This conclusion is still valid if the dimension of the problem increases or if the R^2 value is lowered from 0.995 (in all of Shao's scenarios) to a more realistic value of 0.70.

Resampling methods with compound estimators. Resampling methods are appropriate for compound estimators. The compound estimators identify the correct model most of the time. Results are better with the proposed selection criterion rather than selection by minimum prediction error. A designed experiment tests the effects of outlier density, outlier magnitude, signal-to-noise ratio and resampling method sample sizes. The proposed criterion mostly outperforms or is competitive with the minimum prediction criterion. The signal-to-noise ratio is the most important factor for all methods.

A large portion of the prediction error can be attributed to the large residual values of the outliers. An estimate of prediction error is proposed that weights each observation's squared prediction error by the final weight from a compound estimator. The results are dramatically different from the unweighted estimate of prediction error. There is virtual assurance of selecting the correct model with the proposed criterion and virtual assurance of selecting the largest parameter model with the minimum prediction error criterion. These conclusions hold independent of outlier density, outlier magnitude, or signal-to-noise ratio.

6.4.2 Contributions

- An improved variable selection criterion for bootstrap and cross-validation estimates of prediction error in OLS regression.

- Reliable variable selection is possible in OLS with cross-validation and bootstrap methods if the proposed criterion is used.

- The proposed criterion and resampling methods are recommended for variable selection with compound estimators.

- A weighted estimate of prediction error combined with the proposed criterion is highly effective for variable selection. This method is also robust across a variety of outlier scenarios.

6.4.3 Future Research

This research has demonstrated that resampling methods are appropriate for the variable selection problem in robust regression. A finite sample performance study that compares analytical variable selection procedures from the asymptotic estimates of the covariance matrix to the resampling methods would be useful. Additionally, there are other bootstrap methods proposed that may provide better results. One promising method is the wild bootstrap (Mammen, 1992) that is appropriate for regression models with heteroschedastic errors.

The proposed change in prediction error criterion for variable selection could be improved. This criterion only captures some of the subjectivity in selecting a model and is overly conservative. Improvements are possible by using some measure other than percentage of null model prediction error. A goal programming approach is possible. An opportunity exists to refine the weighted estimate of prediction error. Enhancements to the Ronchetti and Staudte's (1994) robust C_p for resampling could also be considered.

References

Atkinson, A. C. (1994). "Fast very robust methods for the detection of multiple outliers," *Journal of the American Statistical Association*, 89, 1329-1339.

Atkinson, A. C. and Riani, M. (1997). "Bivariate boxplots, multiple outliers, multivariate transformations and discriminant analysis: the 1997 Hunter lecture," *Environmetrics*, 8, 583-602.

Allen, D. M. (1971). "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, 16, 221-227.

Barnett, V. and Lewis, T. (1994). *Outliers in statistical data, 3rd ed*, Wiley: Great Britain.

Barrett, B. E. and Gray, J. B. (1997). "Leverage, residual and interactions diagnostics for subsets of cases in least squares regression," *Computational Statistics & Data Analysis*, 26, 39-52.

Barrett, B. E. and Ling, R. F. (1992). "General classes of influence measures for multivariate regression," *Journal of the American Statistical Association*, 87, 184-191.

Behnken, D. W. and Draper, N. R. (1972). "Residuals and their variance patterns," *Technometrics*, 16, 147-185.

Belsley, D. A., Kuh, E. and Welsch, R. E. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, Wiley: New York, NY.

Breiman, L. (1996). "Bagging predictors," *Machine Learning*, 24, 123-140.

Breiman, L. (1995). "Better subset regression using the nonnegative garrote," *Technometrics*, 37, 373-384.

Breiman, L. (1992). "The little bootstrap and other methods for dimensionality selection in regression: x-fixed prediction error," *Journal of the American Statistical Association*, 87, 738-754.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*, Wadsworth & Brooks/Cole: Pacific Grove, CA.

Breiman, L. and Spector, P. (1992). "Submodel selection and evaluation in regression. The x-random case," *International Statistical Review*, 60, 291-319.

Box, G.E.P. (1953). "Non-normality and tests of variance," *Biometrika*, 40, 318-335.

Burman, R. (1990). "Estimation of optimal transformations using v-fold cross-validation and repeated learning testing methods," *Sankhya, Series A*, 52, 314-345.

Burns, P. J. (1992). "A genetic algorithm for robust regression estimation," *StatSci Technical Note*, Seattle, WA.

Butler, R. W., Davies, P. L., and Jhun, M. (1993). "Asymptotics for the minimum covariance determinant estimator," *Annals of Statistics*, 21, 1385-1400.

Coakley, C. W. and Hettmansperger, T. P. (1993). "A bounded influence, high breakdown, efficient regression estimator," *Journal of the American Statistical Association*, 88, 872-880.

Cook, R. D. (1998). *Regression Graphics*, Wiley: New York.

Cook, R. D. (1979). "Influential observations in linear regression," *Journal of the American Statistical Association*, 74, 169-174.

Cook, R. D. and Hawkins, D. M. (1990). Discussion of Rousseeuw, P. J. and van Zomeren, B. C. "Unmasking multivariate outliers and leverage points," *Journal of the American Statistical Association*, 85, 633-639.

Cook, R. D. and Weisberg, S. (1982). *Residuals and Influence in Regression*, Chapman Hall: New York.

Davies, P. L. (1992). "Asymptotics of Rousseeuw's minimum volume ellipsoid estimator," *Annals of Statistics*, 15, 1269-1292.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap methods and their application*, Cambridge University Press: United Kingdom.

Efron, B. (1983). "Estimating the error rate of a prediction rule: improvement on cross-validation," *Journal of the American Statistical Association*, 76, 312-319.

Efron, B. (1982). "The jackknife, the bootstrap and other resampling plans," *CBMS-NSF Regional Conference Series in Applied Mathematics*, 38, SIAM: Philadelphia.

Efron, B. (1979). "Bootstrap methods: another look at the jackknife," *The Annals of Statistics*, 7, 1-26.

Efron, B. and Tibshirani, R. J. (1997). "Improvements on cross-validation: the .632+ bootstrap method," *Journal of the American Statistical Association*, 92, 548-560.

Efron, B. and Tibshirani, R. J. (1993). *An introduction to the bootstrap*, Chapman & Hall: New York.

Field, C. A. (1997). "Robust regression and small sample confidence intervals," *Journal of Statistical Planning and Inference*, 57, 39-48.

Field, C. A. and Welsh, A. H. (1998). "Robust confidence intervals for regression parameters," *Australia and New Zealand Journal of Statistics*, 40, 55-64.

Gentleman, J. F. and Wilk, M. B. (1975). "Detecting outliers: II supplementing the direct analysis of residuals," *Biometrics*, 31, 387-410.

Gunst, G. F. and Mason, R. L. (1980). *Regression Analysis and its Applications*, Marcel Dekker: New York.

Hadi, A. S. (1994). "A modification of a method for the detection of outliers in multivariate samples," *Journal of the Royal Statistical Society, Series B*, 56, 393-396.

Hadi, A. S. (1992). "Identifying multiple outliers in multivariate data," *Journal of the Royal Statistical Society, Series B*, 54, 761-777.

Hadi, A. S. and Simonoff, J. S. (1993). "Procedures for the identification of multiple outliers in linear models," *Journal of the American Statistical Association*, 88, 1264-1272.

Hadi, A. S. and Simonoff, J. S. (1997), "A more robust outlier identifier for regression data," *Bulletin of the International Statistical Institute*, 281-282.

Hall, P. (1989). "Unusual properties of bootstrap confidence intervals in the regression problem," *Probability Theory and Related Fields*, 70, 247-273.

Hampel, F. R. (1997). "What can the foundations discussion contribute to data analysis? And what may be some future directions in robust methods and data analysis?," *Journal of Statistical Planning and Inference*, 57, 7-19.

Hampel, F. R. (1973). "Robust estimation: a condensed partial survey," *Zeitschrift fur Wahrscheinlichkeitsthiorie und Verwandte Gebiete*, 27, 87-104.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust statistics: the approach based on influence functions*, Wiley: New York.

Handshin, E., Schweppe, F. C., Kohlas, J., and Fiechter, A. (1975). "Bad data analysis for power system state estimation," *IEEE Transactions on Power Apparatus Systems*, PAS-94, 329-337.

Hawkins, D. M. (1994). "The feasible solution algorithm for the minimum covariance determinant estimator in multivariate data," *Computational Statistics and Data Analysis*, 17, 197-210.

Hawkins, D. M. (1993). "A feasible solution algorithm for the minimum volume ellipsoid estimator," *Computational Statistics*, 9, 95-107.

Hawkins, D. M., Bradu, D., and Kass, G. V. (1984). "Location of several outliers in multiple regression data using elemental sets," *Technometrics*, 26, 197-208.

He, X. and Portnoy, S. (1992). "Reweighted LS estimators converge at the same rate as the initial estimator," *The Annals of Statistics*, 20, 2161-2167.

Hertier, S. and Ronchetti, E. (1994). "Robust bounded-influence tests in general parametric models," *Journal of the American Statistical Association*, 89, 897-904.

Hettmansperger, T. P. (1998). Comment on "The goals and strategies of robust methods," by R. R. Wilcox, *British Journal of Mathematical and Statistical Psychology*, 51, 1-39.

Hoaglin, D. C. and Welsch, R. E. (1978). "The hat matrix in regression and ANOVA," *American Statistician*, 32, 17-22.

Huber, P. J. (1981). *Robust statistics*, Wiley: New York.

Huber, P. J. (1973). "Robust regression: asymptotics, conjectures and Monte Carlo simulations," *Annals of Statistics*, 1, 799-821.

Kempthorne, P. J. and Mendel, M. B. (1990). Comment. *Journal of the American Statistical Association*, 85, 647-648.

Kianifard, F. and Swallow, W. (1990). "A Monte Carlo comparison of five procedures for identifying outliers in linear regression," *Communications in Statistics, Part A-Theory and Methods*, 19, 1913-1938.

Kianifard, F. and Swallow, W. (1989). "Using recursive residuals, calculated on adaptively-ordered observations to identify outliers in linear regression," *Biometrics*, 45, 571-585.

Krasker, W. S. and Welsch, R. E. (1982). "Efficient bounded-influence regression estimation," *Journal of the American Statistical Association*, 77, 595-604.

Lee A. H. and Fung, W. K. (1997). "Confirmation of multiple outliers in generalized linear and nonlinear regressions," *Computational Statistics & Data Analysis*, 25, 55-65.

Luceno, A. (1998). "Multiple outliers detection through reweighted least deviances," *Computational Statistics & Data Analysis*, 26, 313-326.

Mammen, E. (1992). *When does the bootstrap work? Asymptotic results and simulations*, Springer-Verlag: New York.

Marasinghe, M. G. (1985). "A multistage procedure for detecting several outliers in linear regression," *Technometrics*, 27, 395-399.

Mallows, C. L. (1975). "On some topics in robustness," unpublished memorandum, Bell Telephone Laboratories: Murray Hill, NJ.

Marazzi, A. (1993). *Algorithms, routines, and S functions for robust statistics*, Wadsworth and Brooks/Cole: Pacific Grove, CA.

Markatou, M. and He, X. (1994). "Bounded influence and high breakdown point testing procedures in linear models," *Journal of the American Statistical Association*, 89, 543-549.

Maronna, R. A. (1976). "Robust M-estimators of multivariate location and scatter," *Annals of Statistics*, 4, 51-67.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models, 2nd ed.*, Chapman and Hall: New York, NY.

Meintanis, S. G. and Donatos, G. S. (1997). "A comparative study of some robust methods for coefficient-estimation in linear regression," *Computational Statistics and Data Analysis*, 23, 525-540

Miller, A. J. (1990). *Subset Selection in Regression*, Chapman and Hall: London.

Montgomery, D. C. and Peck, E. A. (1992). *Introduction to Linear Regression Analysis 2nd ed.*, Wiley: New York.

Paul, S. R. and Fung, K. Y. (1991). "A generalized extreme studentized residual multiple outlier detection procedure in linear regression," *Technometrics*, 33, 339-348.

Quenouille, M. (1949). "Approximate tests of correlation in time series," *Journal of the Royal Statistical Society, Series B*, 11, 18-84.

Rocke, D. M. and Woodruff, D. L. (1997). "Robust estimation of multivariate location and shape," *Journal of Statistical Planning and Inference*, 57, 245-255.

Rocke, D. M. and Woodruff, D. L. (1996). "Identification of outliers in multivariate data," *Journal of the American Statistical Association*, 91, 1047-1061.

Ronchetti, E. (1997). "Robust inference by influence functions," *Journal of Statistical Planning and Inference*, 57, 59-72.

Ronchetti, E. and Staudte, R. G. (1994). "A robust version of Mallows's C_p," *Journal of the American Statistical Association*, 89, 550-559.

Rousseeuw, P. J. (1985). "Multivariate estimation with high breakdown point," *Mathematical Statistics and Applications, Vol B.* eds. W. Grossmann, G. Pflug, I. Vincze and W. Wertz, Reidel: Dordrecht, The Netherlands, 283-297.

Rousseeuw, P. J. (1984). "Least median of squares regression." *Journal of the American Statistical Association*, 79, 871-881.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection,* Wiley-Interscience: New York.

Rousseeuw, P. J. and van Zomeren, B. C. (1990). "Unmasking multivariate outliers and leverage points," *Journal of the American Statistical Association*, 85, 633-639.

Rousseeuw, P. J. and van Zomeren, B. C. (1991). "Robust distances: simulations and cutoff values," *Directions in Robust Statistics and Diagnostics, Part II*, eds. W. Stahel and S. Weisberg, Springer-Verlag: Heidelberg, Germany.

Rousseeuw, P. J. and Yohai, V. J. (1984). "Robust regression by means of S-estimators," *Robust and Nonlinear Time Series Analysis*, eds. J. Franke, W. Hardle, and D. Martin, Springer-Verlag: Heidelberg, Germany.

Ruppert, D. (1992). "Computing S-estimators for regression and multivariate location/dispersion". *Journal of Computational and Graphical Statistics*, 1, 253-270.

Ryan, T. P. (1997). *Modern Regression Methods,* Wiley: New York.

Sebert, D. M. (1996). "Identifying multiple outliers and influential subsets: a clustering approach," unpublished dissertation, Arizona State University, AZ.

Sebert, D. M., Montgomery, D. C. and Rollier, D. (1998). "A clustering algorithm for identifying multiple outliers in linear regression," *Computational Statistics & Data Analysis,* 27, 461-484.

Shao, J. (1996). "Bootstrap model selection," *Journal of the American Statistical Association*, 91, 655-665.

Shao, J. (1993). "Linear model selection by cross-validation," *Journal of the American Statistical Association*, 88, 486-494.

Simonoff, J. S. (1991). "General approaches to stepwise identification of unusual values in data analysis," *Directions in Robust Statistics and Diagnostics, Part II*, eds. W. Stahel and S. Weisberg, Springer-Verlag: Heidelberg, Germany.

Simpson, D. G. and Chang, Y. I. (1997). "Reweighting approximate *GM* estimators: asymptotics and residual-based graphics," *Journal of Statistical Planning and Inference*, 57, 273-293.

Simpson, D. G., Ruppert, D., and Carroll, R. J. (1992). "One-step *GM* estimates and stability of influences in linear regression," *Journal of the American Statistical Association*, 87, 439-450.

Simpson, J. R. (1995). "New methods and comparative evaluations for robust and biased-robust regression estimation," unpublished dissertation, Arizona State University, AZ.

Simpson, J. R. and Montgomery, D. C. (1998a). "The development and evaluation of alternative Generalized-*M* estimation techniques," *Communications in Statistics-Simulation and Computation*, 27, 999-1018

Simpson, J. R. and Montgomery, D. C. (1998b). "A performance –based assessment of robust regression methods," *Communications in Statistics- Simulation and Computation*, 27, 1031-1049.

Simpson, J. R. and Montgomery, D. C. (1998c). "A compound estimator for robust regression," *Naval Research Logistics*, 45, 125-134.

Stone, M. (1974). "Cross-validatory choice and assessment of statistical prediction," *Journal of the Royal Statistical Society, Series B*, 36, 111-133.

Swallow, W. and Kianifard, F. (1996). "Using robust scale estimates in detecting multiple outliers in linear regression," *Biometrics*, 52, 545-556.

Walker, E. (1984). "Influence, collinearity, and robust estimation in regression," unpublished dissertation, Virginia Polytechnic Institute, VA.

Wilcox, R. R. (1998). "The goals and strategies of robust methods," *British Journal of Mathematical and Statistical Psychology*, 51, 1-39.

Wilcox, R. R. (1997). *Introduction to Robust Estimation and Hypothesis Testing*, Academic Press: San Diego, CA.

Wilcox, R. R. (1996a). "Confidence intervals for two robust regression lines with a heteroscedastic error term," *British Journal of Mathematical and Statistical Psychology*, 49, 163-170.

Wilcox, R. R. (1996b). "Confidence intervals for the slope of a regression line when the error term has nonconstant variance," *Computational Statistics and Data Analysis*, 22, 89-98.

Wilcox, R. R. (1994). "Computing confidence intervals for the slope of the biweight midregression and Winsorized regression lines," *British Journal of Mathematical and Statistical Psychology*, 47, 355-372.

Woodruff, D. L. and Rocke, D. M. (1994). "Computable robust estimation of multivariate shape in high dimension using compound estimators," *Journal of the American Statistical Association*, 89, 888-896.

Wu, C. F. J. (1986). "Jackknife, bootstrap, and other resampling methods in regression analysis," *The Annals of Statistics*, 14, 1261-1295.

Yohai, V. J. (1997). "Local and global robustness of regression estimators," *Journal of Statistical Planning and Inference*, 57, 73-93

Yohai, V. J. (1987). "High breakdown-point and high efficiency robust estimates for regression," *The Annals of Statistics,* 15, 642-656.

Yohai,V. J., Stahel, W.A., and Zamar, R.H. (1991). "A procedure for robust estimation and inference in linear regression," *Directions in Robust Statistics and Diagnostics, Part II*, eds. W. Stahel and S. Weisberg, Springer-Verlag: Heidelberg, Germany.

Zhang, P. (1992). "On the distributional properties of model selection criteria," *Journal of the American Statistical Association*, 87, 732-737.

Appendix A

S-Plus Code for Chapter 3 Studies

```
MAKEDATA
#   This script file contains the data generators for the experiments.
#   All clean observations are multivariate normal with mean 7.5 and a
#   standard deviation of 4.0.  These constants are not important to the
#   performance of the procedures.  Some procedures require a column of
#   ones for the constant term (Hadi and Simonoff, Swallow and Kianifard);
#   this is handled internally by the procedures.
#   For multiple point clouds, the regressor levels are perturbed slightly
#   by uniform(0,0.25) to keep them from being a single point mass.
#   The responses are generated by multiplying 5.0 by the level of each
#   predictor value and adding N(0,1) noise and the shift if the cases
#   are regression outliers.
#
#   The subroutine gendata generates the data set with up to two clouds.
#   out1 is the number of outliers in the first cloud, outshft1 is the
#   number of standard deviations to shift the data in X-space, yshift1
#   is the number of standard deviations to shift the response, n is the
#   number of observations, k is the number of regressors and x is the
#   number of regressors that are outlying out of the k.
#   This gendata lets clouds be outlying in fewer than p variables.  The
#   ones not outlying are random.  This is not what gendata2/6 do- they
#   put the other variables at the mean of 7.5.  That configuration is
#   significantly more difficult to detect.
#
gendata<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
{
    {
    outs<-out1+out2 # the total planted outliers
    first<-n-outs+1 # observation number of first planted outlier
    last<-n-outs     # observation number of the last clean case
    kmx<-k-x
    shift1<-7.5 + outshft1*4  # place cloud 1 at this location
    shift2<-7.5 + outshft2*4  # place cloud 2 at this location
#   one<-rep(1,n)                    # some procedures need an intercept
    jin<-matrix(rnorm(last*k,7.5,4.0),ncol=k) # predictors for clean cases
    yin<-apply(5*jin,1,sum) + matrix(rnorm(last),ncol=1)
    yin<-matrix(yin,ncol=1)          # clean response values
    if (k == x)                      # if outlying in all variables
        {
        j1<-matrix(shift1+runif(out1*k,0.0,0.25),ncol=k) # cloud 1 x values
        j2<-matrix(shift2+runif(out2*k,0.0,0.25),ncol=k) # cloud 2 x values
        }
    else
        {
        jout1<-matrix(shift1+runif(out1*x,0.0,0.25),ncol=x)
# outlying subset in cloud
        jin1<-matrix(rnorm(out1*kmx,7.5,4.0),ncol=kmx) # inlying vars in cloud
        j1<-cbind(jout1,jin1)
        jout2<-matrix(shift2+runif(out2*x,0.0,0.25),ncol=x)
        jin2<-matrix(rnorm(out2*kmx,7.5,4),ncol = kmx)
        j2<-cbind(jout2,jin2)
        } #endelse
    x<-rbind(jin,j1,j2) # the x values
#   x<-cbind(one,x)  # if you need the intercept
    x<-as.matrix(x)
    y1<-apply(5*j1,1,sum)+ yshift1 # responses for the first cloud
    y1<-matrix(y1,ncol=1)
    y2<-apply(5*j2,1,sum)+yshift2  # responses for the second cloud
    y2<-matrix(y2,ncol=1)
    y<-rbind(yin,y1,y2)
    y<-matrix(y,ncol=1)
    }
```

```
    return(x,y)
}
#
#   This gendata is for n=40, k=2 and places outliers at a specific place for
#   each of the two variables e.g. 2 sigma for x1 and -2 sigma for x2
#   for up to two clouds.  This is used to keep the responses for the
#   outliers from being y space outliers.
#
gendata2<-function(out1,out2,outshft11,outshft12,outshft21,outshft22,yshift1
    ,yshift2,n,k,x)
{
    {
    outs<-out1+out2 # the total planted outliers
    first<-n-outs+1
    last<-n-outs
    shift11<-7.5 + outshft11*4
    shift12<-7.5 + outshft12*4
    shift21<-7.5 + outshft21*4
    shift22<-7.5 + outshft22*4
#   one<-rep(1,n)
    jin<-matrix(rnorm(last*2,7.5,4.0),ncol=2)
    yin<-apply(5*jin,1,sum) + matrix(rnorm(last),ncol=1)
    yin<-matrix(yin,ncol=1)
    jout11<-matrix(shift11+runif(out1,0.0,0.25),ncol=1)
    jout12<-matrix(shift12+runif(out1,0.0,0.25),ncol=1)
    j1<-cbind(jout11,jout12)
    jout21<-matrix(shift21+runif(out2,0.0,0.25),ncol=1)
    jout22<-matrix(shift22+runif(out2,0.0,0.25),ncol=1)
    j2<-cbind(jout21,jout22)
    x<-rbind(jin,j1,j2) # the x values
#   x<-cbind(one,x)  # intercept
    x<-as.matrix(x)
    y1<-apply(5*j1,1,sum)+ yshift1
    y1<-matrix(y1,ncol=1)
    y2<-apply(5*j2,1,sum)+yshift2
    y2<-matrix(y2,ncol=1)
    y<-rbind(yin,y1,y2)
    y<-matrix(y,ncol=1)
    }
    return(x,y)
}
#
#   The function gendata6 does the same as gendata2 except for k = 6 variables.
#   The specific level for each of the 6 variables can be set in each cloud.
#
gendata6<-function(out1,out2,outshft11,outshft12,outshft13,outshft14,
    outshft15,outshft16,outshft21,outshft22,outshft23,outshft24,outshft25,
    outshft26,yshift1,yshift2,n,k,x)
{
    {
    outs<-out1+out2 # the total planted outliers
    first<-n-outs+1
    last<-n-outs
    shift11<-7.5 + outshft11*4
    shift12<-7.5 + outshft12*4
    shift13<-7.5 + outshft13*4
    shift14<-7.5 + outshft14*4
    shift15<-7.5 + outshft15*4
    shift16<-7.5 + outshft16*4
    shift21<-7.5 + outshft21*4
    shift22<-7.5 + outshft22*4
    shift23<-7.5 + outshft23*4
    shift24<-7.5 + outshft24*4
```

```
        shift25<-7.5 + outshft25*4
        shift26<-7.5 + outshft26*4
#       one<-rep(1,n)
        jin<-matrix(rnorm(last*6,7.5,4.0),ncol=6)
        yin<-apply(5*jin,1,sum) + matrix(rnorm(last),ncol=1)
        yin<-matrix(yin,ncol=1)
        jout11<-matrix(shift11+runif(out1,0.0,0.25),ncol=1)
        jout12<-matrix(shift12+runif(out1,0.0,0.25),ncol=1)
        jout13<-matrix(shift13+runif(out1,0.0,0.25),ncol=1)
        jout14<-matrix(shift14+runif(out1,0.0,0.25),ncol=1)
        jout15<-matrix(shift15+runif(out1,0.0,0.25),ncol=1)
        jout16<-matrix(shift16+runif(out1,0.0,0.25),ncol=1)
        j1<-cbind(jout11,jout12,jout13,jout14,jout15,jout16)
        jout21<-matrix(shift21+runif(out2,0.0,0.25),ncol=1)
        jout22<-matrix(shift22+runif(out2,0.0,0.25),ncol=1)
        jout23<-matrix(shift23+runif(out2,0.0,0.25),ncol=1)
        jout24<-matrix(shift24+runif(out2,0.0,0.25),ncol=1)
        jout25<-matrix(shift25+runif(out2,0.0,0.25),ncol=1)
        jout26<-matrix(shift26+runif(out2,0.0,0.25),ncol=1)
        j2<-cbind(jout21,jout22,jout23,jout24,jout25,jout26)
        x<-rbind(jin,j1,j2) # the x values
#       x<-cbind(one,x)
        x<-as.matrix(x)
        y1<-apply(5*j1,1,sum)+ yshift1
        y1<-matrix(y1,ncol=1)
        y2<-apply(5*j2,1,sum)+yshift2
        y2<-matrix(y2,ncol=1)
        y<-rbind(yin,y1,y2)
        y<-matrix(y,ncol=1)
        }
        return(x,y)
}
#
#    This data generation function generates a single outlying cloud at
#    a random location in the interior of X space found by using the median
#    of the last three clean observations in each variable.   The parameters
#    outshft1,outshft2, yshift2 and x are not used.
#
gendatamed1<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
{
        {
        outs<-out1+out2
        last<-n-outs
        first<-n-outs+1
        lastm2<-last-2
        xin<-matrix(rnorm(last*k,7.5,4.0),ncol=k)
        xmed<-apply(xin[lastm2:last,],2,median)
        temp<-matrix(rnorm(outs*k,0,.05),ncol=k)
        xmedm<-xmed+temp
        xmedm<-matrix(xmedm,ncol=k,byrow=T)
        x<-rbind(xin,xmedm)
        yin<-apply(5*xin,1,sum) + matrix(rnorm(last),ncol=1)
        yout<-apply(5*xmedm,1,sum)+ matrix(rnorm(outs)+yshift1,ncol=1)
        y<-rbind(yin,yout)
        }
        return(x,y,xmed)
}
#
#    The function gendatamed2 does the same thing as gendatamed1 except for 2
#    clouds.   The second cloud is located at the median of the first three
#    clean observations
#
```

```
gendatamed2<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
{
     {
     outs<-out1+out2
     last<-n-outs
     first<-n-outs+1
     lastm2<-last-2
#    one<-rep(1,n)
     xin<-matrix(rnorm(last*k,7.5,4.0),ncol=k)
     xmed1<-apply(xin[lastm2:last,],2,median)
     temp<-matrix(rnorm(out1*k,0,.05),ncol=k)
     xmedm1<-xmed1+temp
     xmedm1<-matrix(xmedm1,ncol=k,byrow=T)
     xmed2<-apply(xin[1:3,],2,median)
     temp<-matrix(rnorm(out2*k,0,.05),ncol=k)
     xmedm2<-xmed2+temp
     xmedm2<-matrix(xmedm2,ncol=k,byrow=T)
     x<-rbind(xin,xmedm1,xmedm2)
#    x<-cbind(one,x)
     yin<-apply(5*xin,1,sum) + matrix(rnorm(last),ncol=1)
     yout1<-apply(5*xmedm1,1,sum)+matrix(rnorm(out1)+yshift1,ncol=1)
     yout2<-apply(5*xmedm2,1,sum)+matrix(rnorm(out2)+yshift2,ncol=1)
     y<-rbind(yin,yout1,yout2)
     }
     return(x,y,xmed1,xmed2)
}
#
#   This function genrand generates regression outliers randomly
#   in x-space.  The parameters outshft1, outshft2, and yshft2 are
#   not used.  These outliers are not in multiple point clouds.
#
genrand<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
{
     {
     outs<-out1+out2
     first<-n-outs+1
     last<-n-outs
     one<-rep(1,n)
     x<-matrix(rnorm(k*n,7.5,4.0),ncol=k)
     yin<-apply(5*x[1:last,],1,sum) + matrix(rnorm(last),ncol=1)
     yin<-matrix(yin,ncol=1)
#    x<-cbind(one,x)
     x<-as.matrix(x)
     yout<-apply(5*x[first:n,],1,sum)+ yshift1
     yout<-as.matrix(yout,ncol=1)
     y<-rbind(yin,yout)
     y<-matrix(y,ncol=1)
     }
     return(x,y)
}
#
#   The function genmix generates low leverage regression outliers (first
#   outliers specified) at random locations in X-space and a cloud
#   of high leverage regression outliers (second outliers specified).
#   The outliers may be unusual in any number of variables in the cloud.
#   The parameter outshft1 is not used since the first set of outliers are
#   random.
#
genmix<-function(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
{
     {
     outs<-out1+out2
     kmx<-k-x
```

```
        first<-n-outs+1
        last<-n-outs
        firststop<-n-out2
        second<-firststop+1
        shift1<-7.5 + outshft1*4
        shift2<-7.5 + outshft2*4
#       one<-rep(1,n)
        xin<-matrix(rnorm(k*firststop,7.5,4.0),ncol=k)
        if (k!=x){
            xcloudout<-matrix(shift2+runif(out2*x,0.0,0.25),ncol=x)
            xcloudin<-matrix(rnorm(kmx*out2,7.5,4.0),ncol=kmx)
            xcloud<-as.matrix(cbind(xcloudout,xcloudin))
                    }
        else       {
            xcloud<-matrix(shift2+runif(out2*k,0.0,0.25),ncol=k)
                    }
        x<-rbind(xin,xcloud)
        x<-as.matrix(x)
#       x<-as.matrix(cbind(one,x))
        yin<-apply(5*x[1:last,],1,sum) + matrix(rnorm(last),ncol=1)
        yin<-matrix(yin,ncol=1)
        yout1<-as.matrix(apply(5*x[first:firststop,],1,sum)+ yshift1,ncol=1)
        yout2<-as.matrix(apply(5*x[second:n,],1,sum)+ yshift2,ncol=1)
        y<-as.matrix(rbind(yin,yout1,yout2),ncol=1)
        }
        return(x,y)
}
#
#    The function gendata4 generates 4 multiple point clouds that may
#    be outlying in a subset of the k variables.  All outlying variables
#    must be at the same level like in "gendata".
#
gendata4<-function(out1,out2,out3,out4,outshft1,outshft2,outshft3,
    outshft4,yshift1,yshift2,yshift3,yshift4,n,k,x)
{
        {
        outs<-out1+out2+out3+out4 # the total planted outliers
        first<-n-outs+1
        last<-n-outs
        kmx<-k-x
        shift1<-7.5 + outshft1*4
        shift2<-7.5 + outshft2*4
        shift3<-7.5 + outshft3*4
        shift4<-7.5 + outshft4*4
#       one<-rep(1,n)
        jin<-matrix(rnorm(last*k,7.5,4.0),ncol=k)
        yin<-apply(5*jin,1,sum) + matrix(rnorm(last),ncol=1)
        yin<-matrix(yin,ncol=1)
        if (k == x)
            {
            j1<-matrix(shift1+runif(out1*k,0.0,0.25),ncol=k)
            j2<-matrix(shift2+runif(out2*k,0.0,0.25),ncol=k)
            j3<-matrix(shift3+runif(out3*k,0.0,0.25),ncol=k)
            j4<-matrix(shift4+runif(out4*k,0.0,0.25),ncol=k)
            }   # end if
        else
            {
            jout1<-matrix(shift1+runif(out1*x,0.0,0.25),ncol=x)
            jin1<-matrix(rnorm(out1*kmx,7.5,4.0),ncol=kmx)
            j1<-cbind(jout1,jin1)
            jout2<-matrix(shift2+runif(out2*x,0.0,0.25),ncol=x)
            jin2<-matrix(rnorm(out2*kmx,7.5,4),ncol = kmx)
            j2<-cbind(jout2,jin2)
```

```
            jout3<-matrix(shift3+runif(out3*x,0.0,0.25),ncol=x)
            jin3<-matrix(rnorm(out3*kmx,7.5,4.0),ncol=kmx)
            j3<-cbind(jout3,jin3)
            jout4<-matrix(shift4+runif(out4*x,0.0,0.25),ncol=x)
            jin4<-matrix(rnorm(out4*kmx,7.5,4),ncol = kmx)
            j4<-cbind(jout4,jin4)
            } #endelse
        x<-rbind(jin,j1,j2,j3,j4) # the x values
#       x<-cbind(one,x)
        x<-as.matrix(x)
        y1<-apply(5*j1,1,sum)+ yshift1
        y1<-matrix(y1,ncol=1)
        y2<-apply(5*j2,1,sum)+yshift2
        y2<-matrix(y2,ncol=1)
        y3<-apply(5*j3,1,sum)+ yshift3
        y3<-matrix(y3,ncol=1)
        y4<-apply(5*j4,1,sum)+yshift4
        y4<-matrix(y4,ncol=1)
        y<-rbind(yin,y1,y2,y3,y4)
        y<-matrix(y,ncol=1)
        }
        return(x,y)
}
#
j<-gendata(3,3,4,5,5,5,60,6,3)
j


SEBERT
#   This S-Plus code implements the Sebert et al. (1998) procedure
#   to identify multiple outliers in datasets.  This code is significantly
#   different from that in Sebert (1996) in order to take advantage of
#   some recent developments in the language and use standard structures
#   across the outlier detection procedures.
#
#   The subroutine resids.func returns the scaled predicted and residual values
#   from OLS regression.
#
resids.func<-function(x,y)
{
    {
        e<-lsfit(x,y)$residuals
        yhat<-y-e
        data<-cbind(yhat,e)
        scaledata<-scale(data)
    }
return(scaledata,e)
}
#
#   The subroutine claster does a single linkage cluster analysis on the scaled
#   predicted and residual values.  The purpose is to identify the clean group
#   of observations and declare all others as candidate outliers.  The clusters
#   are separated by cutting the tree on Mojenas' distance.
#
claster.func<-function(data)
{
    {
        h2<-hclust(dist(data,metric="euclidean"),method="connected")
        maxheight<-h2$height[length(h2$height)]
        meanheight<-mean(h2$height)
        stdheights<-sqrt(var(h2$height))
        mojenas<-meanheight+1.25*stdheights
#   In practice this never occurs, but just in case Mojenas height is
#   greater than the maxheight, cut the tree at maxheight.
```

```
        if(maxheight<=mojenas)
            clustergroups<-cutree(h2,h=maxheight-.01)
        else if(maxheight>mojenas)
            {clustergroups<-cutree(h2,h=mojenas)}
#   Of all the groups formed, the group number of the median observation should
#   be that of the clean subset.
        cleanid<-median(clustergroups)
        outlier<-ifelse(clustergroups==cleanid,0,1)
    }
    return(clustergroups,outlier)
}
#
#   The prog.sim subroutine simulates the procedure for N replications.
#   This determines the percent of outliers detected and average false
#   alarm rate.  The set.seed(i) is required to have common random
#   numbers between the different procedures so the exact same data sets
#   are used to compare the methods.
#
prog.sim<-function(N,out1,out2,shiftx1, shiftx2, shifty1, shifty2,n ,k,x)
{
    {
    outs<-out1+out2  # total outliers
    first<-n-outs+1  # first planted outlying obs #
    last<-n-outs     # last clean obs #
    plant<-0
    false<-0
    i<-1
    while(i<=N){
        set.seed(i)
        cat("iteration ",i," ")
#   Choose any data generating function from makedata.SSC.  Note changes may
#   be required in the prog.sim arguments depending on the selected data set.
        data<-gendata(out1,out2,shiftx1,shiftx2,shifty1,shifty2,n,k,x)
#   generate predicted and residual values.
        predres<-resids.func(data$x,data$y)
        detect.outs<-claster.func(predres$scaledata)
#   determine number of planted outliers detected in this run and add to
#   sum from all previous runs.
        plant<-plant + sum(detect.outs$outlier[first:n])
#   determine false alarms for this run and add to sum of previous runs
        false<-false + sum(detect.outs$outlier[1:last])
        i<-i+1
}
#   from the experiment, the total probability a planted outlier is detected
#   (pp) and the probability a clean observation is classified an outlier.
    pp<-plant/(N*outs)
    po<-false/(N*last)
    }
    return(data,pp,po)
}
Aj<-prog.sim(5,6,6,2,2,5,5,60,6,3)
Aj
```

SWALLOW and KIANIFARD
```
#   This S-PLUS program implements the Swallow and Kianifard multiple outlier
#   detection procedure in Biometrics, 52, pp. 545-556.  It uses MAD and
#   interquartile range as robust estimates of scale.  Outward stepping
#   recursive residuals used to determine outlier status.
#
#   The function sk.madir computes the mean absolute deviation (MAD) and
#   interquartile range (IR) for a clean simulated set of data.
#   It is called in sk.corfact to find correction factors.
#   S-PLUS MAD uses the constant 1.4 consistency in Normal Distribution
```

```
#
sk.madir<-function(n,k)
{
    {
    obs<-n*k
    x<-matrix(rnorm(obs,7.5,4),nrow=n,ncol=k)
    yhat<-NULL
    res<-NULL
    temp<-NULL
    y<-apply(5*x,1,sum) + matrix(rnorm(n),ncol=1)
    olsfit<-lsfit(x,y)
    res<-olsfit$resid
    medresq<-quantile(res,0.50)
    temp<-abs(res-medresq)
    madev<-quantile(temp,0.50)
    ir<-quantile(res,.75)-quantile(res,.25)
}
return(x,y,ir,madev)
}
#
#    Function sk.corfact determines the correction factor for the MAD and IR
#    estimates of scale. This generates Table 1 on page 548.   The MAD
#    correction factors are very close to published values, the IR factors
#    differ (e.g. for n=25, 1.2541 vs published 1.369 and for
#    n = 50, 1.2899 vs 1.363).   For n=60, k=6 use 1.2436 for IR and 0.629
#    for MAD.   For n = 40 and k = 2 use 1.2711 for IR and 0.6452 for MAD
#
sk.corfact<-function(N,n,k)
{
    {
    iqv<-NULL
    madv<-NULL
#   N is the number of simulations (5000) and we create a vector of IR and
#   MAD scale estimates.  The mean of these vectors is the correction factor.
    for (i in 1:N)
        {
        dat<-sk.madir(n,k)
        iqv[i]<-dat$ir
        madv[i]<-dat$madev
        }
    corfir<-mean(iqv)
    corfmad<-mean(madv)
}
return(corfir,corfmad)
}
#
#   This function sk.initial returns the initial clean set of ordered
# observations by externally studentized residual.
#
sk.initial<-function(x,y)
    {
        {
        x<-as.matrix(x)
        y<-as.matrix(y)
        id<-NULL
        vecone<-NULL
        n<-nrow(x)
        k<-ncol(x)
        z<-matrix(0,n,k+3)
        vecone<-rep(1,n)      # vector of 1s
        id<-1:n               # vector identifying observation num
#   We do not have first column of 1s in X
        olsfit<-lsfit(x,y,intercept=TRUE)
```

```
            infl<-ls.diag(olsfit)  # gives access to hat diagonals
            studres<-abs(infl$stud.res) #note this is internally studentized
            temp<-cbind(x,y,studres,id)
            z<-temp[order(temp[,k+2]),]
            z<-as.matrix(cbind(1,z))
# this is the matrix of x,y sorted by |studentized residual|.  Note Z has
# an initial column of 1's.
        }
            return(z,temp) # will not run if only return one value, temp not used
    }
#   Function sk.recursive returns the recursive residuals.  This is not
#   the most efficient code since an updating formula as in Kianifard
#   and Swallow, 1990 could be used.
#
sk.recursive<-function(z)
{
    {
        z<-as.matrix(z)
        k<-ncol(z)-3      # here k = p
        n<-nrow(z)
        nm1<-n-1
        kp1<-k+1
        kp2<-k+2
        corfact<-if(k==7) 0.629 else 0.645
        w<-NULL
        temp<-NULL
        recurres<-NULL
        tswir<-NULL
        tswmad<-NULL
        scaledres<-NULL
#   The i loop goes over the i observations and sequentially adds a clean obs
        for (i in kp1:nm1){
            cleanrows<-i
#   partition our ordered z matrix into clean subset
            cleanx<-z[1:cleanrows,1:k]
            cleany<-z[1:cleanrows,k+1]
            cleanx<-as.matrix(cleanx)
            cleany<-as.matrix(cleany)
            cp1<-cleanrows+1
#   do least squares fit on the clean subset
            olsfit<-lsfit(cleanx,cleany,intercept=FALSE)
            infl<-ls.diag(olsfit)
#   we will use the clean covariance matrix (unscaled by sigma) to determine
#   unscaled prediction error for the potential outliers
            varcov<-infl$cov.unscaled
            varcov<-as.matrix(varcov)
#   This computes equation 3.2 for the recursive residual
                fitted<-sum(olsfit$coef[1:k]*z[cp1,1:k])
                num<-(z[cp1,k+1]-fitted)
                    denom<-sqrt(1+(z[cp1,1:k]%*%varcov%*%z[cp1,1:k]))
                w[cp1]<-num/denom
    }#end i
            recurres<-w[kp2:n]
#   The following computes the test statistics for each observation by using
#   the absolute value of the recursive residual divided by the robust estimate
#   of scale (IR-sigmair or MAD-sigmamad)
#           sigmair<-(quantile(recurres,.75)-quantile(recurres,.25))/1.369
            medresq<-quantile(recurres,0.50)
            temp<-abs(recurres-medresq)
            madev<-quantile(temp,0.50)
            sigmamad<-madev/corfact
#           tswir<-abs(recurres/sigmair)
            tswmad<-abs(recurres/sigmamad)
```

```
#   This set of code tests the distribution of the OLS residuals from the
#   same set of generated data.
          usualols<-lsfit(z[,1:k],z[,k+1],intercept=FALSE)
          usualinfl<-ls.diag(usualols)
          scaledres<-abs(usualols$resid/usualinfl$std.dev)
}
return(tswir,tswmad,scaledres,corfact)
}
#
#   The function sk.studtized returns the test statistics for the studentized
#   residuals rather than the recursive residuals.
#
sk.studtized<-function(z)
{
    {
    z<-as.matrix(z)
    n<-nrow(z)
    k<-ncol(z)-3 # here,  k = p
    vecone<-NULL
    temp<-NULL
    tsstudir<-NULL
    tsstudmad<-NULL
    vecone<-rep(1,n)
    usualols<-lsfit(z[,1:k],z[,k+1],intercept=FALSE)
    usualinfl<-ls.diag(usualols)
    res<-usualols$resid
    sigmair<-(quantile(res,.75)-quantile(res,.25))/1.369
    medresq<-quantile(res,0.50)
    temp<-abs(res-medresq)
    madev<-quantile(temp,0.50)
    sigmamad<-madev/0.639
#   studentized resid = ei/sigmahat(1-hii)^.5
    tsstudir<-res/(sigmair*(sqrt(vecone-usualinfl$hat)))
    tsstudmad<-res/(sigmamad*(sqrt(vecone-usualinfl$hat)))
    }
    return(tsstudmad,tsstudir)
}
#
#   The function sk.critval finds the critical values for the test statistics
#   from simulation.  The procedure is 1. generate clean data (e.g. mv normal)
#   for n=40, 60 etc observations.  2.  find the recursive residuals and the
#   studentized residuals.  3.  find estimates of sigma from IR and MAD-- if
#   using recursive residuals then IR and MAD are on recursive residuals but
#   for studentized residuals, then use IR and MAD on OLS residuals.  4.
#   Do this 5000 times so have 5000 x 25 matrix of test statistics.
#   5.  Find quantiles-note for recursive residual quantiles use 1 sided
#   but use 2 sided for studentized (e.g. for alpha=.05, use 97.5 quantile
#   for Ri and 95 quantile for wi).  This generates table 2  page 550.  MAD
#   has consistent results with table 2.  IR with Studentized residuals
#   deviates most from table 2.
#   For n = 40,  k = 2,  95% is 2.0835, 97.5% is 2.4597 and 99% is 2.8688
#   For n = 60,  k = 6,  95% is 2.053, 97.5% is 2.380 and 99% is 2.797
#
sk.critval<-function(N,n,k)
{
    {
    obs<-n*k
    numrr<-n-k-1   #number of recursive residuals =n-p-1
    tswir<-matrix(0,nrow=N,ncol=numrr)
    tswmad<-matrix(0,nrow=N,ncol=numrr)
    tssir<-matrix(0,nrow=N,ncol=n)
    tssmad<-matrix(0,nrow=N,ncol=n)
    usual<-matrix(0,nrow=N,ncol=n)
```

```
    for (i in 1:N)
        {
            cat("iteration ",i," ")
        x<-matrix(rnorm(obs,7.5,4),nrow=n,ncol=k)
        y<-5*apply(x,1,sum)+rnorm(nrow(x),0,1)
        oaks<-initial.ar(x,y)
        oaksw<-recursive(oaks$z)
        tswir[i,]<-oaksw$tswir
        tswmad[i,]<-oaksw$tswmad
        oakss<-studtized(oaks$z)
        tssir[i,]<-oakss$tsstudir
        tssmad[i,]<-oakss$tsstudmad
        usual[i,]<-oaksw$scaledres
        } # end i
    }
    return(tswir,tswmad,tssir,tssmad,usual)
}
#   To get the critical value, take the appropriate quantile of the large
#   matrix of residuals returned.
#   j<-sk.critval(5000,60,6)
#   j
#   critval.mad<-quantile(j$tswmad,0.975)
#
#   Function sk.ps is the program simulation to determine the detection
#   and false alarm probabilities.  N is the number of replications the rest
#   of the parameters are to generate the data (no col of 1's needed).
#   Of the four possibilities, we consider only the recursive residuals
#   (not studentized) and using the MAD estimate (not IR) of scale.
#
sk.ps<-function(N,out1,out2,xshift1,xshift2,yshift1,yshift2,n,k,x)
{
    {
    teststats<-NULL
    ppl<-NULL
    id<-NULL
    plantdet<-NULL
    pplant<-0.0
    pfalse<-0.0
    outs<-out1+out2                # total outliers
    first<-n-outs+1                # the id of the first planted outlier
    last<-n-outs                   # the id of the last clean observation
    kp3<-k+3                       # determine how large to make initial subset
    critval<-if(k==6) 2.380 else 2.460
    for (i in 1:N)
        {
        cat("iteration ",i,"   ")
        set.seed(i)
        data<-gendata(out1,out2,xshift1,xshift2,yshift1,yshift2,n,k,x)
        sortdata<-sk.initial(data$x,data$y) # ordered by studentized resid
        teststats<-sk.recursive(sortdata$z) # finds recursive residuals
        ppl<-ifelse(teststats$tswmad > critval,1.0,0.0) #exceeds crit val
        idcol<-ncol(sortdata$z)             # observation number location
        id<-sortdata$z[kp3:n,idcol]         # respective observation vector
        plantdet<-ifelse(ppl==0.0 ,0,ifelse(id>last,1,0)) # if detect planted
#                                            outlier then =1 else =0
        false<-ifelse(ppl==0.0,0,ifelse(id>last,0,1)) # here we've exceeded
#                        the critical value but it is not a planted outlier
        pplant<-pplant+sum(plantdet)        # counter for planted outliers
        pfalse<-pfalse+sum(false)           # counter for false alarms
        }
    pp<-pplant/(N*outs)           # probability of detecting planted outlier
    po<-pfalse/(N*(n-outs))        # probability of false alarm
    }
```

```
return(data,pp,po,critval)
}
j<-sk.ps(500,6,6,5,5,10,10,60,6,3)
j


PENA AND YOHAI
#       This program implements the procedure from Pena and Yohai, JRSS (B)
#       , 1995 to detect influential subsets in regression.   The crux of
#       the procedure evaluates the eigenstructure of the influence matrix.
#       The function inflmatrix creates the influence matrix M and outputs the
#       eigenvectors of this matrix.   The computational version given in the
#       equation in section 4 is used as we assume n>>p
#
inflmatrix<-function(x,y)
{
    {
    x<-as.matrix(x)
    y<-as.matrix(y)
    n<-nrow(x)
    p<-ncol(x)
    res<-matrix(0,nrow=n,ncol=1)
    hat<-NULL
    vecone<-rep(1,n)
    olsfit<-lsfit(x,y,intercept=FALSE)
    infl<-ls.diag(olsfit)
    res<-olsfit$resid
    E<-diag(res,nrow=n,ncol=n)# make diagonal matrix of residuals
    E<-as.matrix(E)
    hat<-1/(vecone-infl$hat)    # make diagonal matrix of hii
    D<-diag(hat,nrow=n,ncol=n)
    D<-as.matrix(D)
    temp<-eigen(infl$cov.unscaled) # eigenvectors of (x'x)^-1
#       note that the eigenvectors differ often signficantly
#       if we compute (x'x)^-1 directly (not from ls.diag) unless
#       we specify "digits" to be sufficiently large.
    B<-temp$vectors
    B<-as.matrix(B)
    L<-diag(sqrt(temp$values),nrow=p,ncol=p)
    L<-as.matrix(L)
    A<-B %*% L
    A<-as.matrix(A)
    EDXA<-E %*% D %*% x %*% A
    scaleit<-1.00000000/(sqrt(p)*infl$std.dev)
    P<-scaleit*EDXA
    PtPeig<-eigen(t(P)%*%P)
    Meigvect<-P %*% PtPeig$vectors
    }
    return(Meigvect,p)
}
#
#   The function AUTOID attaches the observation number associated with the
#   eigenvector. It also sorts the eigenvector and finds conditions for
#   outliers. Input is the eigenvectors from the influence matrix and
#   the critical distance k to declare the outlying set.
#
autoid<-function(M,k)
{
    {
    M<-as.matrix(M)
    n<-nrow(M)
    p<-ncol(M)
#   c1 and c2 are the constants used for breakdown adjustment
    c1<-floor(n/4)
```

```
        c2<-floor(n/4)
#               there really is no reason to have both c1 and c2 since we
#               have no way of knowing if the eigenvector will have negative
#               or positive values for the outliers.
        id<-seq(n)
        ev<-array(0,dim=c(p,n,2)) # initialize the sorted eigenvector
        outa<-array(0,dim=c(p,c1,2)) # initialize the array for a values of 4.1.b
#                       "outa" values go with the positive scores
        outb<-array(0,dim=c(p,c2,2)) # initialize the array for b values in 4.1.b
        a<-matrix(0,nrow<-c1,ncol<-p) # initialize the the values for a
        ida<-matrix(0,nrow<-c1,ncol<-p) # matrix has the observation id for a
        values
        idb<-matrix(0,nrow<-c2,ncol<-p) #   idb and b are outb
        b<-matrix(0,nrow<-c2,ncol<-p)
        temp<-matrix(0,ncol=2,nrow=n)
        seta<-matrix(0,ncol=c1,nrow=p)
#  this is the set of outlying observations from the "a" vector.
        setb<-matrix(0,ncol=c1,nrow=p)
        for (i in 1:p)
            {
            temp<-cbind(M[,i],id)
            ev[i,,]<-temp[order(temp[,1]),]  # now eigenvectors are ordered
            for (j in 1:c2)
                {
#  we need to protect against the situation when the value is very close to 0
#  such as .00027 when we divide so we don't get false alarms for the wrong
#  reason.  ev[3,1,2] means third eigenvector, first row, obs id; for the 3rd
#  dimension if use 1, that is the score.  We assign low scores the median
#  value taking into account if it is the positive or negative score.
                medM<-median(abs(M))
                if(abs(ev[i,(j+1),1])< medM) ev[i,(j+1),1]<--medM
                b[j,i]<-ev[i,j,1]/ev[i,(j+1),1]
                idb[j,i]<-ev[i,j,2]
                idx<-j+(3*c2)
                if(abs(ev[i,(idx-1),1])< medM) ev[i,(idx-1),1]<-medM
                a[j,i]<-ev[i,idx,1]/ev[i,(idx-1),1]
                ida[j,i]<-ev[i,idx,2]
                } # end j
            outa[i,,]<-cbind(a[,i],ida[,i])
            outb[i,,]<-cbind(b[,i],idb[,i])
            } # end i
#  Now we form the set of observations which are outliers.  There are p
#  eigenvectors but we only use c1 of the scores.  The constant k is key here.
#  It measures how large of a difference between two scores has to be before
#  declaring the set outlying.  Simulations show a value of 2.5 is perhaps too
#  small based on the number of false alarms.  The authors suggest step 2
#  (t tests) will correct the false alarm problems.
        for (i in 1:p)
            {
            for (j in 1:c1)
                {
                if (outa[i,j,1]>k) # if ratio of scores for positive scores > k
                    {
                    idx<-j
                    while (idx<=c1)
#  take all observations from the breakpoint up to n as outliers.
                        {
                        seta[i,idx]<-outa[i,idx,2]
                        idx<-idx+1
                        } # endwhile
                    } # end if
                if (outb[i,j,1]>k) # ratio of scores for negative scores
                    {
```

```
                        idx<-j
                        while(idx>0)
# take all observations from the breakpoint back to 1 (the most neg)
                            {
                            setb[i,idx]<-outb[i,idx,2]
                            idx<-idx-1
                            }#endwhile
                        } # endif
                    }#end j
                }#end i
        }
return(outa,seta,outb,setb)
}
#
#   This function simulates the procedure N times for the n observations each
#   run.   An N x n matrix called obs is used to compute the proportion of
#   correctly identified observations since it is known the outliers were
#   planted as the last few cases.
#
#   This is for genrand, gendata, gendatamed2,gendatamed1
prog.sim<-function(N,out1,out2,xshift1,xshift2,yshift1,yshift2,n,k,x)
#   This is for gendata2
#prog.sim<-function(N,out1,out2,xs11,xs12,xs21,xs22,yshift1,yshift2,n,k,x)
#   This is for gendata6
# prog.sim<-
    function(N,out1,out2,xs11,xs12,xs13,xs14,xs15,xs16,xs21,xs22,xs23,xs24,xs25
    ,xs26,yshift1,yshift2,n,k,x)
{
    {
    out<-out1+out2
    firstout<-n-out+1
    lastclean<-n-out
    p<-k+1
    c1<-floor(n/4)
    obs<-matrix(0,nrow=N,ncol=n)
    for (i in 1:N)
        {
        cat("iteration   ",i," ",n," ")
        set.seed(i)
# This generates data from gendatamed1,2, genrand or gendata
        a<-gendata(out1,out2,xshift1,xshift2,yshift1,yshift2,n,k,x)
# This generates data from gendatbig2
#        a<-gendatbig2(out1,out2,xs11,xs12,xs21,xs22,yshift1,yshift2,n,k,x)
# This generates data from gendatbig6
#        a<-
    gendatbig6(out1,out2,xs11,xs12,xs13,xs14,xs15,xs16,xs21,xs22,xs23,xs24,xs25
    ,xs26,yshift1,yshift2,n,k,x)
        a$x<-cbind(1,a$x)
        b<-inflmatrix(a$x,a$y)
        c<-autoid(b$Meigvect,2.5)
#   The following code looks at the observations declared outliers from both
#   set A and set B.  If the observation appears in either set, the obs matrix
#   is assigned a value of 1.  This avoids the double counting an observation
#   that may appear as an outlier from two separate eigenvectors.  This obs
#   matrix can then be used to compute any statistic of interest from the
#   simulation.
        for (j in 1:p)
            {
            for (l in 1:c1)
                {
                if (c$seta[j,l]>0)
                    {
                    temp<-c$outa[j,l,2]
```

```
                                obs[i,temp]<-1
                                } # end if
                        if (c$setb[j,1]>0)
                                {
                                temp<-c$outb[j,1,2]
                                obs[i,temp]<-1
                                } # end if
                        } #end l
                } #end j
        }   # end i
        avg<-apply(obs,2,mean)
#  pp is the percentage of outliers correctly identified while pp is the
#  probability of swamping clean observations.
#
        pp<-mean(avg[firstout:n])
        po<-mean(avg[1:lastclean])
    }
    return(a,pp,po)
}
j<-prog.sim(5,4,4,5,5,5,5,40,2,2)
j


ROUSSEEUW and VAN ZOMEREN
#  This code incorporates the Rousseeuw and van Zomeren (1990) procedure with
#  both rule of thumb/chi square and simulated critical cut off values.
#  The subroutine critvals computes the simulated critical cutoff values
#  for the scaled residuals from the LMS fit.  We generate
#  lots of N * n clean residuals and find the appropriate  percentiles.
#  For n = 40, k = 2 use 3.61 for 98.75% or 3.01 for 97.5%
#  for LMS.  Use the same for n = 60, k = 6 for LMS.  For n = 40, k = 2 use
#  3.38 for 98.75% and 2.87 for 97.5% and for n = 60 k = 6 use 4.11
#  for 98.75% and 3.51 for 97.5% for LTS.
#
critvals<-function(N,n,k)
{
    {
    resmat<-matrix(0,nrow=N,ncol=n)
    for (i in 1:N)
        {
        set.seed(i)
        cat("iteration ", i, " ")
        datapts<-n*k
#  generate clean x matrix for a run multivariate normal(7.5, 4^2)
        x<-matrix(rnorm(datapts,7.5,4),ncol=k)
        y<-apply(5*x,1,sum)
        y<-matrix(y,ncol=1)+matrix(rnorm(n),ncol=1)
#       a<-lmsreg(x,y)
        a<-ltsreg(x,y)
        stdresi<-a$residuals/a$scale
        resmat[i,]<-stdresi
        }
    q95<-quantile(resmat,0.95)
    q975<-quantile(resmat,0.975)
    q9875<-quantile(resmat,0.9875)
    q05<-quantile(resmat,0.05)
    q025<-quantile(resmat,0.025)
    q0125<-quantile(resmat,0.0125)
    }
 return(q95,q975,q9875,q05,q025,q0125)
}
# b4<-critvals(50,60,6)
# b4
#
```

```
#   The subroutine robdist computes the robust distances with MVE estimator
#
robdist<-function(x)
{
    {
        x<-as.matrix(x)
        n<-nrow(x)
        p<-ncol(x)
        transpx<-t(x)
        varcov<-var(x)
        mn<-apply(x,2,mean)
        md<-mahalanobis(x,mn,varcov)
#   This section computes the minimum volume ellipsoid robust distances
        v<-cov.mve(x)
        dmve<-mahalanobis(x,v$center,v$cov)
    }
    return(dmve,md)
}
#
#   The subroutine reglms computes the least median of squares regression
#   residuals and returns two n vectors: 1) chkrot is 1 if the observation
#   residual value exceeds the rule of thumb cutoff else it is 0,
#   2) chksim is 1 if the residual exceeds the simulated cutoff value, 0 o.w.
#
reglms<-function(x,y)
{
    {
        j<-lmsreg(x,y)
#   Need scaled residuals
        stdresi<-abs(j$residuals/j$scale)
        chkrot<-ifelse(stdresi>2.5,1,0)
        chksim<-ifelse(stdresi>3.61,1,0)
    }
    return(chkrot,chksim,stdresi)
}
#
#   The subroutine prog.sim determines the probability the planted outliers
#   are detected and the false alarm probability for various outlier scenarios
#
prog.sim<-function(N,out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x,iter)
{
    {
    outs<-out1+out2 # the total planted outliers
    first<-n-outs+1 # first observation that is a planted outlier
    last<-n-outs    # last clean observation
#   initialize values
    summves<-0  # total detected with simulated R&vZ
    summvefs<-0 # total R&vZ false alarms
    summver<-0  # total detected with original R&vZ
    summvefr<-0 # total false alarms for original R&vZ
#   critical values for the MVE procedure differ with parameters
    chicrit<-if(k==2)7.3984 else 14.45
    mvesimcrit<-if(k==2)9.3225 else 17.935
#   generate data sets
    for (i in 1:N){
        cat("you're on iteration ",i,"   ")
        set.seed(i)
        data<-gendata(out1,out2,outshft1,outshft2,yshift1,yshift2,n,k,x)
        rdist<-robdist(data$x,iter)

#
#   The MVE procedure.  Note the simulated critical value is
#   17.935 for the 97.5% if n = 60 and p = 6 variables.  For n = 40 and
```

```
#  p = 2, then we use the simulated value as 9.9225.  Rousseeuw and
#  and Zomeren recommend using Chi Square with p degrees of freedom.
#  For our case that would be p = 2 degrees of freedom so the
#  For LMS, the recommendation is 2.5 and the simulated value is 3.61
#  (98.75th) to control total experimentwise error to 5%
#
        resout<-reglms(data$x,data$y)
        mvems<-ifelse(rdist$dmve>mvesimcrit,1,0)
        resdiss<-ifelse(mvems+resout$chksim>0,1,0)
        mveouts<-sum(resdiss[first:n])
        summves<-summves + mveouts
        mvefalses<-sum(resdiss[1:last])
        summvefs<-summvefs + mvefalses
#  Rule of thumb critical values from Chi Square are 7.3984 for p = 2
#  and 14.45 for p = 6.  These are the critical values for robust
#  distances based on alpha = 0.025.
        mvemr<-ifelse(rdist$dmve>chicrit,1,0)
        resdisr<-ifelse(mvemr+resout$chkrot>0,1,0)
        mveoutr<-sum(resdisr[first:n])
        summver<-summver + mveoutr
        mvefalser<-sum(resdisr[1:last])
        summvefr<-summvefr + mvefalser
    }   #end for
#  Statistics for all the runs. pp is proportion of planted outliers detected
#  po is the probability clean observations are classified as outliers
        ppmves<-summves/(N*outs)
        pomves<-summvefs/(N*(n-outs))
        ppmver<-summver/(N*outs)
        pomver<-summvefr/(N*(n-outs))
        }
return(data,ppmves,pomves,ppmver,pomver)
}
j<-prog.sim(5,6,6,5,5,9,9,60,6,3,1000)
j


REGRESSION ESTIMATORS
#  This program finds outliers using the rediduals regression
#  estimators.  The first step is finding the critical cutoff
#  value to determine if the observation is an outlier.  Next,
#  the observations are classified for the run and tallied over
#  the number of replications.
#  make sure you load robeth library >library(robeth)
#  The subroutine critval calculates the quantiles of clean data for
#  various n and k.  These are the avg of 2.5th and 97.5th quantiles
#  For M, use 1.85 for both n = 60, k = 6 and n = 40, k = 2
#  For LTS use 3.56 for n = 60 and k = 6 and 2.87 for  n = 40, k = 2
#  For LMS use 3.01 for both sample sizes.
#  For MM use 1.90 for both
#  For Simpson, use 1.981 for both
#  For BM and OLS(Walker GM), use 1.960 for both
#  For CH (Coakley Hettmansperger), use 2.084 for both
#
critval<-function(N,n,k)
{
    {
    values<-matrix(0,nrow=N,ncol=n)
    obs<-n*k
    for (i in 1:N)
        {
        cat("iteration ",i," ")
        set.seed(i)
        x<-matrix(rnorm(obs,7.5,4),ncol=k)
        y<-apply(5*x,1,sum) + matrix(rnorm(n),ncol=1)
```

```
                    #   put in whatever regression estimator you want the quantiles for
          #   in the next line.  chreg, ltsreg, lmsreg, lsfit, myhbhe (MM), rreg (M)
                a<-bmreg(x,y)
                values[i,]<-a$residuals
                } # end for
          q95<-quantile(values,0.95)
          q975<-quantile(values,0.975)
    #   use 98.75 in Rousseeuw and van Zomeron type applications
    #   to keep experimentwise error to a total 5%
          q9875<-quantile(values,0.9875)
          q99<-quantile(values,0.99)
          q05<-quantile(values,0.05)
          q01<-quantile(values,0.01)
          q0125<-quantile(values,0.0125)
          q025<-quantile(values,0.025)
}
return(q95,q975,q9875,q99,q05,q025,q0125,q01)
}
#b<-critval(1000,60,6)
#b
#
#   The function prog.sim generates the datasets and determines the
#   probability the residuals detect the outliers and the false alarm
#   probability.
prog.sim<-function(N,out1,out2,xs1,xs2,yshift1,yshift2,n,k,x)
{
    {
    outs<-out1+out2
    first<-n-outs+1
    last<-n-outs
    sumfalseols<-0
    sumdetectols<-0
    sumfalsebm<-0
    sumdetectbm<-0
    sumfalsech<-0
    sumdetectch<-0
    sumfalsejs<-0
    sumdetectjs<-0
    sumdetectlms<-0
    sumfalselms<-0
    sumdetectlts<-0
    sumfalselts<-0
    sumfalsemm<-0
    sumdetectmm<-0
    sumdetectm<-0
    sumfalsem<-0
#   only the critical value of LTS estimator depends on dimension
    cvlts<-if(n==60) 3.56 else 2.87
    for(i in 1:N)
        {
        set.seed(i)
        cat("iteration ",i, "   ")
        data<-gendata(out1,out2,xs1,xs2,yshift1,yshift2,n,k,x)
        a<-bmreg(data$x,data$y)
        b<-chreg(data$x,data$y)
        c<-lmsreg(data$x,data$y)
        d<-myhbhe(data$x,data$y)
        e<-lsfit(data$x,data$y)
        f<-bijs5sa(data$x,data$y)
        g<-rreg(data$x,data$y)
        h<-ltsreg(data$x,data$y)
#   Bounded influence estimator (Walker)
        outliersbm<-ifelse(abs(a$residuals)>1.96,1,0)
```

```
            falsebm<-sum(outliersbm[1:last])
            sumfalsebm<-sumfalsebm+falsebm
            detectbm<-sum(outliersbm[first:n])
            sumdetectbm<-sumdetectbm+detectbm
#   CH- Coakley Hettmansperger compound estimator
            outliersch<-ifelse(abs(b$residuals)>2.084,1,0)
            falsech<-sum(outliersch[1:last])
            sumfalsech<-sumfalsech+falsech
            detectch<-sum(outliersch[first:n])
            sumdetectch<-sumdetectch+detectch
# OLS
            outliersols<-ifelse(abs(e$residuals)>1.96,1,0)
            falseols<-sum(outliersols[1:last])
            sumfalseols<-sumfalseols+falseols
            detectols<-sum(outliersols[first:n])
            sumdetectols<-sumdetectols+detectols
# LTS
            outlierslts<-ifelse(abs(h$residuals)>cvlts,1,0)
            falselts<-sum(outlierslts[1:last])
            sumfalselts<-sumfalselts+falselts
            detectlts<-sum(outlierslts[first:n])
            sumdetectlts<-sumdetectlts+detectlts
#
# LMS
            outlierslms<-ifelse(abs(c$residuals)>3.01,1,0)
            falselms<-sum(outlierslms[1:last])
            sumfalselms<-sumfalselms+falselms
            detectlms<-sum(outlierslms[first:n])
            sumdetectlms<-sumdetectlms+detectlms
# Simpson and Montgomery estimator
            outliersjs<-ifelse(abs(f$residuals)>1.981,1,0)
            falsejs<-sum(outliersjs[1:last])
            sumfalsejs<-sumfalsejs+falsejs
            detectjs<-sum(outliersjs[first:n])
            sumdetectjs<-sumdetectjs+detectjs
#   M
            outliersm<-ifelse(abs(g$residuals)>1.85,1,0)
            falsem<-sum(outliersm[1:last])
            sumfalsem<-sumfalsem+falsem
            detectm<-sum(outliersm[first:n])
            sumdetectm<-sumdetectm+detectm
#   MM
            outliersmm<-ifelse(abs(d$rs1)>1.90,1,0)
            falsemm<-sum(outliersmm[1:last])
            sumfalsemm<-sumfalsemm+falsemm
            detectmm<-sum(outliersmm[first:n])
            sumdetectmm<-sumdetectmm+detectmm
          } #end for
      }
    ppbm<-sumdetectbm/(N*outs)
    pobm<-sumfalsebm/(N*last)
    ppch<-sumdetectch/(N*outs)
    poch<-sumfalsech/(N*last)
    ppols<-sumdetectols/(N*outs)
    pools<-sumfalseols/(N*last)
    pplts<-sumdetectlts/(N*outs)
    polts<-sumfalselts/(N*last)
    pplms<-sumdetectlms/(N*outs)
    polms<-sumfalselms/(N*last)
    ppjs<-sumdetectjs/(N*outs)
    pojs<-sumfalsejs/(N*last)
    ppm<-sumdetectm/(N*outs)
    pom<-sumfalsem/(N*last)
```

```
    ppmm<-sumdetectmm/(N*outs)
    pomm<-sumfalsemm/(N*last)
    return(data,ppbm,pobm,ppch,poch,ppols,pools,pplts,polts,pplms,polms,ppjs,po
    js,ppm,pom,ppmm,pomm)
}
j<-prog.sim(500,3,3,5,5,5,5,40,2,2)
j
```

Appendix B

S-Plus Code and Data for Chapter 4 Studies

```
#   LEVERAGE STUDY
#   The leverage study evaluates several robust distance measures.  The
#   MVE and MCD estimates of mean and covariance matrix are available internal
#   to S-Plus with the command cov.mve, cov.mcd.  Also built in is the
#   function mahalanobis which can calcute the robust distances if the mean
#   and covariance matrix are supplied.  Distances for M-estimates of
#   covariance are available from the ROBETH library and the Simpson and
#   Montgomery compound estimator code in Ch 5 shows how to do that.  The
#   code for Hadi (1992, 1994)is not shown, but is available from his web
#   site.  The code to implement the C++ version of R&W is shown below.

#   ROCKE AND WOODRUFF PROCEDURE
#
#   The robust distances from Rocke and Woodruff (1996) is written in C++.
#   A callable S+ routine can be formed by creating a dynamic data
#   link in a C++ compiler.  This is not a trivial process.  The dll
#   is called "multoutlier.dll" and is accessed via
#       >dll.load("c:\\mydir\\multoutlier.dll","MultOut","cdecl")
#
#   The function multo actually calls the C++ code.  The input
#   values are the data set x and the number of variables p.  The next
#   5 values in the function are output from the dll.  mn and cov are the
#   robust mean and covariance matrix estimates, dist is the n vector of
#   robust distances, rej is the simulated critical cutoff value and
#   status is an internal report for algorithm function. Note initial
#   values must be input during a function call.
#
#
multo<-function(x, p, mn, cov, dist, rej, status)
   {
    .C("MultOut",
        as.double(x),
        as.double(p),
        as.double(mn),
        as.double(cov),
        as.double(dist),
        as.double(rej),
        as.integer(status))
   }
#
#   The function rocky gets the R&W robust distances and cutoff values.
#   The robust distances from MVE, MCD and the Mahalanobis distance can
#   also be calculated (Ch 4 leverage study) if desired.
rocky<-function(x, iter)
{
    {
        x <- as.matrix(x)
        n <- nrow(x)
        p <- ncol(x)
#   This section computes the Mahalanobis distance
        transpx <- t(x)
#       varcov <- var(x)
#       mn <- apply(x, 2, mean)
#       md <- mahalanobis(x, mn, varcov)
#   This section computes the minimum volume ellipsoid robust distances
#       v<-cov.mve(x)
#       dmve<-mahalanobis(x,v$center,v$cov)
#   This section computes the minimum covariance determinant robust distances
#       cd<-cov.mcd(x)
#       dmcd<-mahalanobis(x,cd$center,cd$cov)
#   The initialize variables for the Rocke and Woodruff algorithm
        status <- 0  #   returns an error code if encountered like singular data
```

```
            rejdis <- 3  # if the robust distance is beyond this, then an outlier
            dist <- rep(0, n)   #   initial robust distance vector
#   These are the inputs to the Rocke and Woodruff procedure.  p is number of
#   variables, n is the number of observations, iter specifies how many
#   iterations are used for the smooth estimators (Tukey Biweight M-estimate)
#   The authors recommend n^2 iterations and this is a sensitive parameter.
#   the next two 0's use the default values for the seed and lamba multiplier
#   0.05 is used as the alpha value for the cutoff value
#   the last two 0's are used for the simulation tolerance and trace options
            parms <- c(p, n, iter, 0, 0, 0.05, 0, 0)
            j <- multo(transpx, parms, mn, varcov, dist, rejdis, status)
            distance <- as.numeric(j[[5]])   # the 5th output is the robust distance
            reject <- as.numeric(j[[6]]) # the sixth is the cutoff value

    }
#   add md, dmve, and dmcd to the return list if desired
    return(distance, reject)

}
#   INITIAL ESTIMATOR STUDY
#   This is the proposed initial estimator in Chapter 4 P1 that uses a
#   high breakdown R&W filter to clear high leverage observations
#   followed by a high breakdown MM estimate to remove the outliers on
#   interior X-space.  The coefficients are estimated with OLS on the
#   observations that remain.
#
P1<-function(x,y,iterat=1500)
{
    {
    x<-as.matrix(x)
    p<-ncol(x)
#   upon entry find the unusual observations in X-space with
#   Rocke and Woodruff procedure
    rwdist<-rocky(x,iterat)
    good<-ifelse(rwdist$distance<rwdist$reject,1,0)
    goodx<-x[good==1,]
    goodxint<-cbind(1,x[good==1,])
    goody<-y[good==1]
#   MM estimator from ROBETH library on low leverage observations
    mm<-myhbhe(goodxint,goody)
#   MM estimator internal to SPLUS (many problems with datasets
#   in Chapter 4)
#   mm<-lmRobMM(goody~goodx,efficiency=.90)
#   The simulated critical value for 2 tailed 95% is 1.90 for both
#   n=60,  k = 6 and n = 40, k = 2.
    cleanx<-goodx[abs(mm$rs1)<1.90,]
    cleany<-goody[abs(mm$rs1)<1.90]
    initest<-lsfit(cleanx,cleany)
#   find out how many observations were used in the OLS fit.
    percentobs<-length(initest$residuals)/length(y)
#   how many observations removed for high leverage.
    pctobsx<-length(goody)/length(y)
    predval<-initest$coef%*%t(xint)
    resids<-y-t(predval)
    medres<- median(abs(resids))
# median of absolute residuals
    scale <- 1.4826 * (1 + (5/(length(y) - p - 1))) * medres
# lms scale estimate
    }
list(robdist=rwdist$distance,reject=rwdist$reject,coef=initest$coef,pctobsx=pc
    tobsx,percentobs=percentobs,scale=scale,residuals=resids)

}
#   The function P2 is the initial estimate formed by clearing high leverage
#   points with Rocke and Woodruff, followed by an MM estimate on the
#   remaining observations
```

```
P2<-function(x,y,iterat=1500)
{
    {
    x<-as.matrix(x)
    p<-ncol(x)
#   Rocke and Woodruff procedure clears high leverage observations
    rwdist<-rocky(x,iterat)
    good<-ifelse(rwdist$distance<rwdist$reject,1,0)
    goodx<-x[good==1,]
    goodxint<-cbind(1,x[good==1,])
    goody<-y[good==1]
#   MM estimator from ROBETH
    mm<-myhbhe(goodxint,goody)
#   Internal Splus MM estimator (had trouble with data sets in Ch 4)
#   mm<-lmRobMM(goody~goodx,efficiency=.90)
    percentobs<-length(goody)/length(y)
    xint<-cbind(1,x)
    predval<-mm$theta1 %*%t(xint)
    resids<-y-t(predval)
    medres<-median(abs(resids))
# median of absolute residuals
    scale <- 1.4826 * (1 + (5/(length(y) - p - 1))) * medres
# lms scale
    }
list(robdist=rwdist$distance,reject=rwdist$reject,coef=mm$coef,percentobs=perc
    entobs,scale=scale,residuals=resids)
}
#   For the P3 initial estimator, substitute the function "sest(x,y)" for
#   "myhbhe(x,y)" in P2 and "coef" instead of "theta1"
#
#   This is the S estimate function using the ROBETH library.
sest<-function(x,y){
    {
        # need column of ones
        x<-as.matrix(x)
        x<-cbind(1,x)
        y<-as.matrix(y)
        np<-ncol(x)
        npp1<-np+1
        dfvals()
        dfrpar(x,'S')
        ribetu(y)
        zr<-hysest(x,y,npp1,iopt=1,intch=1,iseed=5431)
        coef<-zr$theta[1:np]
        smin<-zr$smin
        rs<-zr$rs
        nrep<-zr$nrep
        cov<-zr$cov
        ierr<-zr$ierr
        dfcomn(ipsi=4,xk=1.5477)
        S.w<-Psi(rs/smin)/(rs/smin) #weights
        }
    list(coef=coef,resid=rs,nrep=nrep,smin=smin,cov=cov,ierr=ierr,w=S.w)
}
#   The following is the code for the proposed compound estimator CEP1.  The
#   initial estimate is P1 (OLS estimate after R&W and MM filter).  For CEP2
#   just use P2 instead of P1 in the init argument.  LMS measure of scale
#   and pi weights from the R&W robust distances are used in both estimators.
#   The other components follow that of the Simpson and Montgomery estimator.
#   Do not include column of 1s in for x matrix.
CEP1<-function(x, y, w = rep(1, nrow(x)), int = TRUE, init = P1(x,y),
                    method = wt.bibisquare, wx, iter = 1,
                    acc = 50 * .Machine$single.eps^0.5, test.vec = "resid")
```

```
{
            coefin <-coef <- init$coef
            x <- as.matrix(cbind(1, x))
    if(!missing(wx)) {
        if(length(wx) != nrow(x))
            stop("Length of wx must equal number of observations")
        if(any(wx < 0))
            stop("Negative wx value")
        w <- w * wx
    }
    if(ncol(x) != length(coef))
        stop("Must have same number of initial values as coefficients")
    resid <- init$residuals
#   Determine the tuning constant based on the suggestion of Marazzi and
#   Joss (1993)
    tc_4.685
    xwt_as.matrix(x)
#
#   Scale the distances such that the median distance is unity and all others
#   are a ratio of the R&W distance to the median R&W distance
        rockwood.dis<-init$robdis/median(init$robdis)
        pi<-1/rockwood.dis
#   LMS-estimator scale estimate
    scale <- init$scale
#   IRLS step
    for(iiter in 1:iter) {
            epis_c(resid/(scale*pi))
#   In case the residuals go to zero, keeps the weight = 1 (vs undefined)
            if(any(resid == 0)) {
                for (i in 1:length(y)) {
                    if (resid[i] == 0)
                        w[i] <- 1
                    else
                        w[i] <- method(epis[i],tc)
                }
            }
            else
#   Tukey biweight
                w <- method(epis,tc)
#   if any weights are missing set them to 0.9999 and write them to a file
#           if(any(is.na(w)))
#               break
#
            if(!missing(wx))
                w <- w * wx
            temp <- lsfit(x, y, w, int = FALSE)
            coef <- temp$coef
            resid <- temp$residuals
    }
    if(!missing(wx)) {
        tmp <- (wx != 0)
        w[tmp] <- w[tmp]/wx[tmp]
    }
    list(coef = coef, initialest = coefin, init.pct=init$percentobs,residuals =
    resid, scale = scale, tc = tc, distances = rockwood.dis,
            piweight = pi, erroverpis = epis, w = w, int = int)
}
#   The proposed estimators CEP3 and CEP4 use the Simpson and Montgomery shell
#   and integrate the R&W distances into the code.  CEP4 is the same code
#   except iter=4.
CEP3<-  function(x, y, robdis=rocky(x,1500), w = rep(1, nrow(x)), int = TRUE,
    init = sest(x,y), method = wt.bibisquare, wx, iter = 3, acc = 50 *
    .Machine$single.eps^0.5, test.vec = "resid")
```

```
{
#   rockdis<-rocky(x,2000)
           rockdism<-robdis$distance/median(robdis$distance)
    if(int) {
             coef <- init$coef
             coefin <- coef
       x <- cbind(1, x)
    }
    else {
                 init <- sest(x,y,int=FALSE)
             coefin <- coef   <- init$coef
                  x <- as.matrix(x)
    }
    if(!missing(wx)) {
        if(length(wx) != nrow(x))
            stop("Length of wx must equal number of observations")
        if(any(wx < 0))
            stop("Negative wx value")
        w <- w * wx
    }
    if(ncol(x) != length(coef))
        stop("Must have same number of initial values as coefficients")
    resid <- y - x %*% coef
#   Determine the tuning constant based on the suggestion of Marazzi and
#   Joss (1993)
    tc_4.685
if (int==F)    xwt_as.matrix(cbind(1,x))
      else
    xwt_as.matrix(x)
# Robeth pi weights using the scatter matrix
    dfrpar(xwt, "Kra-Wel")
    # Weights
    z         <- wimedv(xwt)
    z         <- wynalg(xwt, z$a); nitw <- z$nit
#   Scale the distances such that the median distance is unity and all others
#   are a ratio of the actual distance to the median distance
#   If any of the design points are at the design center (z$dist=0)
        if(any(z$dist <= 1)) {
            for (i in 1:length(y))  {
              if (z$dist[i] <= 1)    z$dist[i] <- 1
              }
        }
        z$distm <- z$dist/median(z$dist)
        pi<-1/rockdism
#    pi      <- 1/z$distm
#   S-estimator scale estimate
    scale <- init$smin
    for(iiter in 1:iter) {
        if(scale == 0) {
            convi <- 0
            method.exit <- TRUE
            status <- "could not compute scale of residuals"
        }
        else {
            epis_c(resid/(scale*pi))
#   In case the residuals go to zero, keeps the weight = 1 (vs undefined)
            if(any(resid == 0)) {
                for (i in 1:length(y)) {
                    if (resid[i] == 0)
                        w[i] <- 1
                else
                        w[i] <- method(epis[i],tc)
                }
```

```
            }
            else
                w <- method(epis,tc)
#   if any weights are missing set them to 0.9999 and write them to a file
#           if(any(is.na(w)))
#               break
#
            if(!missing(wx))
                w <- w * wx
            temp <- lsfit(x, y, w, int = FALSE)
            coef <- temp$coef
            resid <- temp$residuals
             }
    }
    if(!missing(wx)) {
        tmp <- (wx != 0)
        w[tmp] <- w[tmp]/wx[tmp]
    }
    list(coef = coef, initialest = coefin, residuals = resid, scale = scale, tc
    = tc, distances = rockdism,
                piweight = pi, erroverpis = epis, w = w, int = int)
}
wt.bibisquare_
#   bounded influence WEIGHT FUNCTION where w(t) = psi (t) / t   and
#   t = e / pi*s The Bisquare psi function
#   user supplied tuning constant
function(u, tc=4.685)
{
    U <- abs(u/tc)
    si <- u*(1 - (u/tc)^2)^2
    si[U > 1] <- 0
    w <- si/u
    w
}
```

Data for example 4.1. The last 12 observations are the outliers.

Obs	X1	X2	X3	X4	X5	X6	Y
1	5.3823301	9.614453	3.7734586	2.521202	6.4050258	7.3501984	104.32255
2	7.6399143	2.761681	6.1712453	10.984803	5.6254019	12.9707766	137.11706
3	13.0370290	11.331870	9.4999389	7.167426	6.7438372	3.7328693	155.51306
4	10.7727443	8.287176	9.1914531	11.028556	8.9037412	5.5707952	161.92864
5	2.7299034	-1.738653	3.3689351	7.553015	5.7227006	9.6624944	83.52686
6	10.4810030	2.690387	22.4556538	13.196081	5.7760354	3.3430037	172.90492
7	6.8460480	5.545758	7.5374386	7.799578	9.5848066	9.3440537	139.17059
8	8.6021905	-1.574534	11.9032765	3.324497	10.4984211	10.5185110	128.91487
9	4.4957646	3.790461	0.3168387	5.724236	0.6394165	10.8383479	78.02622
10	10.4108857	4.461523	7.5206740	4.838975	13.1717577	10.1787169	152.02609
11	3.1328522	8.138250	2.9910044	6.240829	8.4822331	6.4009793	106.45168
12	9.4262638	7.357037	3.8303179	7.417321	12.2748921	7.8987294	144.52452
13	12.8340702	7.881631	4.5699923	8.136197	7.9018917	7.6343578	145.19953
14	10.1217210	1.611387	5.7285887	5.484543	4.5462922	2.4431619	89.67091
15	8.9949684	6.869055	6.6792119	2.068845	9.8762037	3.9012816	114.03957
16	12.2033635	10.081425	4.2619827	10.952933	0.2118726	10.0866595	142.99390
17	12.1672918	1.413197	4.2331657	13.452437	3.5444161	6.9164573	124.84412
18	-0.4786179	1.525314	9.6412884	4.616496	13.4350978	13.1610760	127.09250
19	14.6193903	9.283246	6.6081707	9.794120	8.4357788	2.1251370	151.81062
20	10.6389691	5.949906	12.7882345	15.382925	5.1049394	8.8794022	175.27584
21	11.2357283	5.370224	6.5255288	8.013817	6.0347466	7.9635401	134.30437
22	10.7607252	11.602980	16.5012272	6.577516	6.1512523	4.4999197	168.84336
23	10.1715367	9.667104	9.0716426	4.011757	8.7646003	1.2395447	130.11057
24	5.7416098	11.669826	5.3067366	7.778743	3.3951572	12.5589753	139.58273
25	5.4380052	12.543017	3.7095641	3.875349	2.3731280	2.3656449	91.13215
26	6.7779650	6.755774	12.9642941	11.490602	4.3153579	10.6817471	160.91060
27	13.9724213	8.935317	6.9040932	7.952625	16.5185430	8.9170345	189.07445
28	6.9133736	9.966795	12.4596465	15.719241	6.0990198	9.7320189	181.27074
29	6.4451249	10.713168	10.1851605	8.677254	5.6937397	6.0968341	143.43590
30	4.4175348	3.451324	2.5703336	6.685750	9.4718800	3.7627264	91.61120
31	11.1833606	2.740354	3.3807407	1.463547	7.8313128	3.1243734	88.75565
32	9.5206835	11.347720	13.5530517	7.391091	8.4444563	6.3873713	169.54299
33	9.2972117	2.755905	5.7720189	8.710240	14.0915529	6.1806423	140.38910
34	8.0820229	10.581643	2.2820935	6.751160	10.8112616	4.6771552	130.98061
35	6.6545338	5.953882	11.7974144	1.632838	12.4576491	4.2776184	127.09196
36	9.7789906	4.450119	8.0398496	7.723112	2.1111587	6.4538818	112.99237
37	3.7122426	8.483071	9.6422982	10.898808	0.4416167	7.4082651	123.70719
38	2.9175638	9.570676	8.6705725	5.872688	6.6981518	6.9191682	120.60296
39	7.1712089	8.301989	9.2621336	7.337809	6.0533967	7.6988907	136.22295
40	7.9885509	7.922253	4.3438396	11.867781	11.9704074	6.5147955	150.71991
41	10.7296348	3.940886	9.6727416	4.490182	2.4137826	5.0522809	109.20729
42	3.3579211	11.545688	15.5056035	10.811871	15.8786297	18.3340809	225.63244
43	6.9905263	5.888745	6.3327476	12.437475	1.0124723	13.4010089	137.23044
44	4.2694368	7.624350	11.4710680	2.795954	8.1789689	7.4919210	126.20630
45	4.2436118	8.247273	7.9382609	10.072959	8.3573557	5.0815099	131.69331
46	13.3403445	6.637980	7.3312521	7.299546	9.7278347	5.5564581	147.82376
47	7.4250368	-3.658136	7.2793227	7.416180	7.8721623	0.1856133	79.36270
48	10.3821435	6.405248	7.2371363	7.276544	-2.8245757	6.8317375	105.12162
49	27.6440714	27.668316	10.5576269	2.885532	10.5387302	1.3897659	252.05213
50	27.5673301	27.656082	10.7283546	7.581958	9.8268777	6.8145485	280.52545
51	27.5595650	27.502891	12.5874381	4.058082	1.9078757	2.8545462	239.41120
52	27.7100060	27.537986	8.6447890	9.936585	1.0224269	-3.3482467	224.51064
53	27.6890687	27.648046	4.0823853	6.285512	6.5082089	6.5091707	246.16718
54	27.6885707	27.649998	8.8239860	2.580198	4.1724778	5.0386919	237.86177
55	29.6444015	29.607418	10.7805351	2.198492	6.7131428	7.5417817	269.95731
56	29.6764561	29.740360	8.9489976	15.255337	9.8877447	-0.3237614	290.05540
57	29.6907864	29.606927	4.5866859	3.414381	6.3052529	8.3756654	256.43910
58	29.5212998	29.631905	1.6110610	2.583814	9.9902880	9.9422569	260.34187
59	29.6534562	29.579621	9.7561766	5.063578	11.6280968	8.6546168	293.50664
60	29.6933737	29.513834	6.5407833	4.360302	5.9092306	12.7901649	276.92307

Monte Carlo simulation data for Figure 4.1 showing the area of coverage for the various compound estimators. Each cell gives the proportion of times out of 50 replicates that the procedure assigned a standardized residual value of 2.5 or greater.

δ_L	δ_R	C&H	S&M	CE P1P2	CE P3	CE P4
0	3	0.70	0.24	0.12	0.10	0.10
0	4	0.98	1.00	0.94	0.92	0.92
0	≥5	1.00	1.00	1.00	1.00	1.00
1	3	0.12	0.00	0.00	0.00	0.00
1	4	0.74	0.06	0.16	0.02	0.02
1	5	0.98	0.92	0.44	0.58	0.44
1	6	1.00	1.00	0.82	0.88	0.80
1	7	1.00	1.00	1.00	1.00	0.96
2	4	0.04	0.08	0.00	0.00	0.00
2	5	0.16	0.36	0.04	0.04	0.02
2	6	0.38	0.72	0.20	0.38	0.28
2	7	0.72	0.74	0.56	0.62	0.48
2	8	0.84	0.88	0.70	0.84	0.76
2	9	0.90	0.92	0.98	0.98	0.92
3	5	0.00	0.08	0.14	0.04	0.10
3	6	0.14	0.14	0.28	0.14	0.18
3	7	0.18	0.24	0.18	0.22	0.14
3	8	0.50	0.44	0.56	0.46	0.54
3	9	0.50	0.74	0.74	0.72	0.70
3	10	0.50	0.84	0.82	0.84	0.80
4	3	0.00	0.00	0.50	0.02	0.12
4	4	0.00	0.00	0.78	0.10	0.24
4	5	0.00	0.00	0.80	0.08	0.42
4	6	0.00	0.04	0.80	0.30	0.60
4	7	0.00	0.02	0.80	0.34	0.72
4	8	0.00	0.14	0.78	0.50	0.76
4	9	0.00	0.16	0.84	0.56	0.76
4	10	0.00	0.38	0.96	0.82	0.98
5	3	0.00	0.00	0.56	0.04	0.22
5	4	0.00	0.00	0.74	0.08	0.34
5	5	0.00	0.00	0.96	0.22	0.68
5	6	0.00	0.00	1.00	0.36	0.82
5	7	0.00	0.02	0.96	0.52	0.86
5	8	0.00	0.00	0.96	0.70	0.92
5	9	0.00	0.10	0.96	0.72	0.80
6	3	0.00	0.00	0.07	0.06	0.40
6	4	0.00	0.00	0.84	0.28	0.56
6	5	0.00	0.00	0.92	0.32	0.58
6	6	0.00	0.00	1.00	0.48	0.84
6	7	0.00	0.00	1.00	0.70	0.94
6	8	0.00	0.00	1.00	0.84	0.96

Data for Figure 4.1 (cont)

7	3	0.00	0.00	0.62	0.12	0.32
7	4	0.00	0.00	0.80	0.26	0.54
7	5	0.00	0.00	0.84	0.30	0.36
7	6	0.00	0.00	0.96	0.44	0.72
7	7	0.00	0.00	0.98	0.68	0.88
7	8	0.00	0.00	1.00	0.86	1.00
8	3	0.00	0.00	0.54	0.28	0.40
8	4	0.00	0.00	0.84	0.30	0.56
8	5	0.00	0.00	0.84	0.44	0.64
8	6	0.00	0.00	0.94	0.58	0.78
8	7	0.00	0.00	1.00	0.66	0.86
8	8	0.00	0.00	1.00	0.86	1.00
9	3	0.00	0.00	0.54	0.22	0.40
9	4	0.00	0.00	0.84	0.46	0.56
9	5	0.00	0.00	0.84	0.40	0.70
9	6	0.00	0.00	0.94	0.66	0.86
9	7	0.00	0.00	1.00	0.78	0.92
9	8	0.00	0.00	1.00	0.86	0.96
10	3	0.00	0.00	0.68	0.36	0.54
10	4	0.00	0.00	0.76	0.50	0.62
10	5	0.00	0.00	0.86	0.54	0.70
10	6	0.00	0.00	0.90	0.64	0.76
10	7	0.00	0.00	0.98	0.82	0.92
10	8	0.00	0.00	0.96	0.82	0.96

Appendix C

S-Plus Code for Chapter 5 Studies

```
#   The function bijs5 is the abbreviated version of the Simpson and
#   Montgomery (1998)estimator.  It provides only coefficient estimates,
#   residuals and final weights for computational considerations.
bijs5<-function(x, y, w = rep(1, nrow(x)), int = TRUE, init = fastsest(x, y),
    method = wt.bibisquare, wx, iter = 1,
acc = 50 * .Machine$single.eps^0.5, test.vec = "resid")
{
    {
        coef <- init$coef
        x <- cbind(1, x)  #  w <- w * wx
        resid <- y - x %*% coef   #  Determine the tuning constant based on
    the suggestion of Marazzi and Joss (1993)
        tc <- 4.685
        xwt <- as.matrix(x)   # Robeth pi weights using the scatter matrix
        dfrpar(xwt, "Kra-Wel")    # Weights
        z <- wimedv(xwt)
        z <- wynalg(xwt, z$a)
        nitw <- z$nit    #  Scale the distances such that the median distance
    is unity and all others are a ratio of the
#   actual distance to the median distance
        z$distm <- z$dist/median(z$dist)
        pi <- 1/z$distm  #  S-estimator scale estimate
        scale <- init$smin
        epis <- c(resid/(scale * pi))
        w <- method(epis, tc)
        temp <- lsfit(x, y, w, int = FALSE)
        coef <- temp$coef
        resid <- temp$residuals
        if(!missing(wx)) {
            tmp <- (wx != 0)
            w[tmp] <- w[tmp]/wx[tmp]
        }
    }
    list(coef = coef, residuals = resid, weight=w)
}
#   The function fastsest is the initial S estimator for the abbreviated
#   version of the Simpson and Montgomery compound estimator.
fastsest<-function(x, y)
{
    {
# need column of ones
        x <- as.matrix(x)
        x <- cbind(1, x)
        y <- as.matrix(y)
        np <- ncol(x)
        npp1 <- np + 1
        dfvals()
        dfrpar(x, "S")
        ribetu(y)
        zr <- hysest(x, y, npp1, iopt = 1, intch = 1, iseed = 5431)
        coef <- zr$theta[1:np]
        smin <- zr$smin
        rs <- zr$rs
        nrep <- zr$nrep
        cov <- zr$cov
        ierr <- zr$ierr
        dfcomn(ipsi = 4, xk = 1.5477)
        S.w <- Psi(rs/smin)/(rs/smin)  #weights
    }
    list(coef = coef, smin = smin)
}
#   The function gendatagm generates the Gunst and Mason (1980) data set
#   used in the Shao (1993, 1996) studies.  The regressor levels are
```

```
#   always the Gunst and Mason data while the responses are generated
#   from the known beta vector + N(0,sigma) as in Shao. The input parameters
#   are seedy for the random number seed and sigma to control the signal-to
#   noise ratio.   Shao uses sigma = 1.0.
gendatagm<-function(seedy,sigma)
{
    {
        var2 <- c(0.36, 1.2, 0.06, 0.16, 0.01, 0.02, 0.56, 0.98, 0.32, 0.01,
    0.15, 0.24, 0.11, 0.08, 0.61, 0.03, 0.06, 0.02, 0.04, 0,
            0.09, 0.02, 0.02, 0.05, 0.11, 0.18, 0.04, 0.85, 0.17, 0.08, 0.38,
    0.11, 0.39, 0.43, 0.57, 0.13, 0.04, 0.13, 0.2, 0.07)
        var3 <- c(0.53, 2.52, 0.09, 0.41, 0.02, 0.07, 0.62, 1.06, 0.2, 0,
    0.25, 0.28, 0.35, 0.13, 0.85, 0.03, 0.11, 0.08, 0.24, 0.02,
            0.18, 0.16, 0.11, 0.24, 0.39, 0.11, 0.09, 1.33, 0.32, 0.12, 0.18,
    0.13, 0.38, 0.46, 1.16, 0.03, 0.05, 0.18, 0.95, 0.06
            )
        var4 <- c(1.06, 5.74, 0.27, 0.83, 0.07, 0.07, 2.12, 2.89, 0.76, 0.07,
    0.5, 0.59, 0.4, 0.28, 0.49, 0.23, 0.5, 0.25, 0.08, 0.04,
            0.59, 0.24, 0.21, 0.43, 0.29, 0.43, 0.23, 2.7, 0.66, 0.49, 0.49,
    0.18, 0.99, 1.47, 1.82, 0.08, 0.14, 0.28, 0.41, 0.18)
        var5 <- c(0.5326, 3.6183, 0.2594, 1.0346, 0.0381, 0.344, 1.4459,
    4.0182, 0.46, 0.154, 0.6516, 0.0611, 0.1922, 0.0931, 0.0538,
            0.0199, 0.0419, 0.1093, 0.0328, 0.0797, 0.1855, 0.1572, 0.0998,
    0.2804, 0.2879, 0.681, 0.3242, 2.6013, 0.4469, 0.2436,
            0.44, 0.3351, 1.3979, 2.0138, 1.9356, 0.105, 0.2207, 0.018,
    0.1017, 0.0962)
#   The following line augments the design matrix with 5 noise variables
#   if desired
#        noise<-matrix(rnorm(200),nrow=40)
#        x <- matrix(cbind(1, var4, var5, var2, var3,noise), ncol = 10)
#   Otherwise, the original data...
        x <- matrix(cbind(1, var4, var5, var2, var3), ncol = 5)
#   beta is the known generating vector.  Note the order of the variables
#   is changed to reflect Shao.
        beta <- matrix(c(2, 4, 8, 0, 0), ncol = 1)
        set.seed(seedy)
        error <- matrix(rnorm(40, 0, sigma), ncol = 1)
        y <- x %*% beta + error
        x <- x[, -1]
    }
    return(x, y)
}
#   The function gendatagmc generates the modified Gunst and Mason data
#   used in Chapter 5 with 10% outliers planted.  The example in Chapter
#   1 uses 129 as seedy.  The variable sigma determines how much noise
#   is added and if it is >= 5, no estimator works.
gendatagmc<-function(seedy, sigma)
{
    {
        j <- gendatagm(seedy, sigma)
        beta <- matrix(c(2, 4, 8, 0, 0), ncol = 1)
        j$x <- cbind(1, j$x)
#   Plant outliers a distance of 10 sigma away
        j$y[8]  <- j$x[8,  ] %*% beta + 10*sigma
        j$y[15] <- j$x[15, ] %*% beta + 10*sigma
        j$y[28] <- j$x[28, ] %*% beta + 10*sigma
        j$y[39] <- j$x[39, ] %*% beta + 10*sigma
        y <- j$y
        x <- j$x[, -1]
    }
    return(x, y)
}
#   The function gendatanormc generates N(0,1) regressors and calculates
```

```
#   the response by multiplying by beta and adding N(0,sigma) noise.
#   The last 5 of 40 observations are outliers.  The last 3 are also high
#   leverage. The inputs are the seed, sigma (amount of noise added to
#   response), dis (residual and leverage magnitude for the outliers)
#   out (number of outliers)
gendatanormc<-function(seedy, out, dis, sigma)
{
            {
first<-40-out+1 #find the first observation to make an outlier
set.seed(seedy)
x <- matrix(rnorm(160), nrow = 40)
                beta <- matrix(c(2, 4, 8, 0, 0), ncol = 1)
                x <- cbind(1, x)
                x[37:40, 2:5] <- x[37:40, 2:5] + dis
                y <- x %*% beta + rnorm(40, 0, sigma)
                y[first:40] <- y[first:40] + dis
                x <- x[, -1]
        }
        return(x, y)
}
# auxiliary function matmin finds the element of a vector that is the minimum
# and assigns it a value of 1 while all others are 0.
matmin<-function(x)
{
    minx <- min(x)
    minvec <- ifelse(x == minx, 1, 0)
    return(minvec)
}
#   auxiliary function matmax finds the maximum element in a vector
matmax<-function(x)
{
    maxx <- max(x)
    maxvec <- ifelse(x == maxx, 1, 0)
    return(maxvec)
}
# auxiliary function sstman finds the total sums of squares/n
sstman<-function(isub, y)
{
    sst <- sum((y[isub] - mean(y[isub]))^2)/length(y[isub])
    return(sst)
}
#   auxiliary function costcol finds the prediction error for a model.
#   vector y is the original data and vector x is the predicted.
costcol<-function(x, y)
{
    x <- matrix(x, ncol = 1)
    y <- matrix(y, ncol = 1)
    j <- sum((y - x)^2)/length(y)
    return(j)
}
#   The function regboot1 performs regression using x[isub] to predict y[isub]
#   isub is a vector of length n,
#   a bootstrap sample from the sequence of integers
#   1, 2, 3, ..., n
#
#   This function is used by other functions when computing
#   bootstrap estimates.  x is regressors (without intercept), regfun is
#   the regression estimator (lsfit, bijs5), m tells how large the bootstrap
#   sample should be (full sample use m=0).  The matrix of coefficients
#   from the B bootstrap samples and the bootstrap prediction error for each
#   bootstrap sample (for the bias correction) are returned.
#   Also returns the weighted estimate of prediction error if use a robust
#   estimator, if use lsfit, then comment out the last wtderr<- line.
```

```
regboot1<-function(isub, x, y, regfun, m)
{
    {
        wtderr<-NULL
        nmm <- nrow(x) - m
        xmat <- matrix(x[isub,  ], nrow = nmm, ncol(x))
        regboot <- regfun(xmat, y[isub])
        coefficients <- matrix(regboot$coef, ncol = 1)
        xmat <- cbind(1, xmat)
#  bspe finds the prediction error for this bootstrap sample using
#  the bootstrap response values.  This is needed for the unbiased estimate.
        bspe <- sum((regboot$residuals)^2)/length(y[isub])
#  wtderr weights the  prediction error with the compound estimators final
#  weights.
        wtderr<-sum(((regboot$residuals)^2)*regboot$weight)/length(y[isub])
    }
    list(coef = coefficients, booterr = bspe, wtderr=wtderr)
}
#  The function willbs executes the bootstrap and returns the average
#  prediction error if m != 0 or the bias corrected prediction error
#   if m = 0.  Note that x does not have a column of 1s.
willbs<-function(x, y, data, regfun = bijs5, nboot = 100, m)
{
    {
        x <- as.matrix(x)
        p <- ncol(x) + 1
        y <- matrix(y, ncol = 1)
        bvec <- apply(data, 1, regboot1, x, y, regfun, m)
#  bvec is the p+1 by nboot matrix. The first row
#  contains the bootstrap intercepts, the second row
#  contains the bootstrap values for first predictor, etc.
        bootpe <- NULL
        bootwtderr<-NULL
        coef <- matrix(0, ncol = nboot, nrow = p)
#  this piece of inefficient code extracts the coefs and bootstrap
#  resubstitution error for each bootstrap sample.
        for(i in 1:nboot) {
                coef[, i] <- bvec[[i]]$coef
                bootpe[i] <- bvec[[i]]$booterr
                bootwtderr[i]<-bvec[[i]]$wtderr
            }
#  The n by nboot matrix of predicted values using the bootstrap coefficients
#  contained in the matrix coef and the real x's
        pred <- cbind(1, x) %*% coef
#  avg prediction error vector of length nboot if use the
#  bootstrap predictions and the observed y's
        apevec <- apply(pred, 2, costcol, y)
# contains the vector of average prediction error
#  If use m = 0 (full bootstrap sample size) then need the unbiased estimate
#  of prediction error.  resub is the usual resubstitution error using only
#  original data.
        resub <- sum((y - cbind(1, x) %*% bijs5(x, y)$coef)^2)/length(y)
        apevec.unbias <- (apevec - bootpe) + resub
        apevec.wtd<-bootwtderr
    }
    return(apevec, apevec.unbias,apevec.wtd)
}
#  The function win.pure takes a vector of prediction errors and returns the
#  dimension of the best model.  It is not necessarily the model with the
#  lowest prediction error.  Input the constant (const) for minimum change
#  in prediction error required before going to the next higher dimension.
win.pure<-function(prederr,const=.025)
{
```

```
#   Find the critical value for minimum change in prediction error from the
#   model with one less dimension before a variable
#   can be added.  This is similar to the CART impurity criteria for splitting
#   nodes Brieman et al. (1984).
        min.purity <- const * prederr[1]  #  Determine the change in impurities
        delta <- matrix(0, nrow = 1, ncol = 5)
#   change ncol in above line for 10 variable models
#   delta[1,1] is made very large to offset the vector by 1 to account for
#   the intercept.
        delta[1, 1]  <- 100000
        delta[1, 2]  <- prederr[1] - prederr[2]
        delta[1, 3]  <- prederr[2] - prederr[3]
        delta[1, 4]  <- prederr[3] - prederr[4]
        delta[1, 5]  <- prederr[4] - prederr[5]
#        delta[1, 6]  <- prederr[5] - prederr[6]
#        delta[1, 7]  <- prederr[6] - prederr[7]
#        delta[1, 8]  <- prederr[7] - prederr[8]
#        delta[1, 9]  <- prederr[8] - prederr[9]
#        delta[1, 10] <- prederr[9] - prederr[10]

#   the pure vector determines if the variable should be added
        pure <- ifelse(delta < min.purity, 1, 0)
#   create an index of to choose the j-parameter model
        j <- rep(1:5, 1)
#   the best model is the last time the change in impurity is > crit val
        winner <- max(j[pure == 0])
#   additional check if the change in prediction
#   error occurs for high dim models then make sure it is lower than
#   prediction error for the true 5 variable model.  This really should
#   be done several times.
#        if(prederr[winner] > prederr[5]) {
#                pure[winner] <- 1
#                winner <- max(j[pure == 0])
#        }
        return(winner)
}
#   shaosimgmn executes the entire bootstrap simulation by inputting the number
#   of replications(iter), the number of outliers in the sample (out), the
#   magnitude of the outliers (dis), the noise in the sample, NID (0,sigma)
#   the number of observations to remove from the full sample for the bootstrap
#   sample (m), and a seed.
#
shaosimgmr<-function(iter, out,dis,sigma,m,nboot, seedy)
{
        {
#   initialize the values of the matrices that store the number of times
#   each model is selected
        cumpct.shao <- matrix(0, nrow = 1, ncol = 5)
        cumpct.jw <- matrix(0, nrow = 1, ncol = 5)
        cumwin.shao <- matrix(0, nrow = 1, ncol = 5)
        cumwin.jw <- matrix(0, nrow = 1, ncol = 5)
        cumpct.shaowt <- matrix(0, nrow = 1, ncol = 5)
        cumpct.jwwt <- matrix(0, nrow = 1, ncol = 5)
        cumwin.shaowt <- matrix(0, nrow = 1, ncol = 5)
        cumwin.jwwt <- matrix(0, nrow = 1, ncol = 5)
#   Replications
                for(i in 1:iter) {
                cat("iter ", i," ")
                    seeder <- seedy + i
                    j <- gendatanormc(seeder, out,dis, sigma)
# data is the bootstrap resample matrix for all nboot samples.
                    p <- ncol(j$x) + 1
                    nmm <- length(j$y) - m
```

```
                        data <- matrix(sample(length(j$y), size = nmm * nboot,
                                replace = T), nrow = nboot)
#   wbs1 calculates the nboot prediction errors for a 1 variable model
                        wbs1 <- willbs(j$x[, 1], j$y, data, nboot = nboot, m = m)
                        wbs2 <- willbs(j$x[, 1:2], j$y, data, nboot = nboot, m = m)
                        wbs3 <- willbs(j$x[, 1:3], j$y, data, nboot = nboot, m = m)
                        wbs4 <- willbs(j$x[, 1:4], j$y, data, nboot = nboot, m = m)
#                       wbs5 <- willbs(j$x[, 1:5], j$y, data, nboot = nboot, m = m)
#                       wbs6 <- willbs(j$x[, 1:6], j$y, data, nboot = nboot, m = m)
#                       wbs7 <- willbs(j$x[, 1:7], j$y, data, nboot = nboot, m = m)
#                       wbs8 <- willbs(j$x[, 1:8], j$y, data, nboot = nboot, m = m)
#                       wbs9 <- willbs(j$x[, 1:9], j$y, data, nboot = nboot, m = m)
#   Now that we have the contending models avg pred error for all nboot fits
#   put them in a 5 by nboot matrix to find out the lowest prediction error of
#   the 5 contenders in each of the nboot trials.  We first find the model with
#   no predictors as the total sum of squares.
                        SST <- matrix(apply(data, 1, sstman, j$y), nrow = 1)
#   Use the unbiased prediction error if m = 0 (bootstrap sample size  = n)
#                       if(m != 0){
#   For the outlier study (tab 5.13), we do not want to use the bias
#   correction for the full sample so we'll bypass it since m will never = 3.
                        if(m != 3){
                            j <- matrix(rbind(SST, wbs1$apevec, wbs2$apevec,
                                    wbs3$apevec, wbs4$apevec), nrow = 5)
                            jwt <- matrix(rbind(SST, wbs1$apevec.wtd,
        wbs2$apevec.wtd, wbs3$apevec.wtd, wbs4$apevec.wtd), nrow = 5)}
                            else j <- matrix(rbind(SST, wbs1$apevec.unbias, wbs2$
        apevec.unbias, wbs3$apevec.unbias, wbs4$apevec.unbias, wbs5$apevec.unbias,
        wbs6$apevec.unbias, wbs7$apevec.unbias, wbs8$apevec.unbias,
        wbs9$apevec.unbias), nrow = 10)
                            wbs1 <- NULL
                            wbs2 <- NULL
                            wbs3 <- NULL
                            wbs4 <- NULL
                            wbs5 <- NULL
                            wbs6 <- NULL
                            wbs7 <- NULL
                            wbs8 <- NULL
                            wbs9 <- NULL
#   pct.shao finds the selection percentage of the nboot samples using
#   the minimum prediction error criteria.
                        pct.shao <- apply(apply(j, 2, matmin), 1, sum)
                        pct.shaowt <- apply(apply(jwt, 2, matmin), 1, sum)
#       cumpct.shao tallies this percentage over the iter iterations
                        cumpct.shao <- cumpct.shao + pct.shao
                        cumpct.shaowt <- cumpct.shaowt + pct.shaowt
#   win.shao selects the model with the lowest average prediction error
#   across the nboot samples.
                        win.shao <- matmin(apply(j, 1, mean))
                        win.shaowt <- matmin(apply(jwt, 1, mean))
#   cumwin.shao tallies the winners up across the iter iterations
                        cumwin.shao <- cumwin.shao + win.shao
                        cumwin.shaowt <- cumwin.shaowt + win.shaowt
#   pct.jw is the percentage of times the model is selected in the
#   nboot samples in purity metric rather than absolute minimum aggregate
#   prediciton error.
                        jw <- apply(j, 2, win.pure,const=.005)
                        jwwt <- apply(jwt, 2, win.pure,const=.001)
                        pct.jw <- c(sum(ifelse(jw == 1, 1, 0)), sum(ifelse(jw ==
                                2, 1, 0)), sum(ifelse(jw == 3, 1, 0)), sum(
                                ifelse(jw == 4, 1, 0)), sum(ifelse(jw == 5, 1,
                                0)))
```

```
                              pct.jwwt <- c(sum(ifelse(jwwt == 1, 1, 0)),
sum(ifelse(jwwt ==2, 1, 0)), sum(ifelse(jwwt == 3, 1, 0)), sum(ifelse(jwwt ==
4, 1, 0)), sum(ifelse(jwwt == 5, 1,0)))
                              cumpct.jw <- cumpct.jw + pct.jw
                              cumpct.jwwt <- cumpct.jwwt + pct.jwwt
#   win.jw finds the best model based on the average of prediction error
#   from the nboot samples with the minimum change in prediction error metric.
                              win.jw <- win.pure(apply(j, 1, mean),const=.005)
                              idx <- rep(0, 5)
                              idx[win.jw] <- 1
                              cumwin.jw <- cumwin.jw + idx      #  clear the arrays
                              win.jwwt <- win.pure(apply(jwt, 1, mean),const=.001)
                              idx <- rep(0, 5)
                              idx[win.jwwt] <- 1
                              cumwin.jwwt <- cumwin.jwwt + idx
                              j<-NULL
                              jwt<-NULL
                  }

            }
            return(cumpct.shao, cumwin.shao, cumpct.jw, cumwin.jw, cumpct.shaowt,
      cumwin.shaowt, cumpct.jwwt, cumwin.jwwt, j, jwt)
}
#   The function cvpress calculates the leave-one-out estimate of
#   prediction error by performing n regressions.  It also provides the
#   weighted avg prediction error.
cvpress<-function(x, y, method = bijs5)
{
            {
                  set.seed(129)
                  x <- as.matrix(x)
                  n <- nrow(x)
                  y <- as.matrix(y, ncol = 1)
                  xint <- matrix(cbind(1, x), nrow = n)
                  cvpred <- matrix(0,nrow=n,ncol=1)
                  prederr<-matrix(0,nrow=n,ncol=1)
#   loop through all n observations and leave one out each time
                  for(i in 1:n) {
                        cvreg <- method(x[ - i,   ], y[ - i])
                        predvals <- xint %*% cvreg$coef
                        cvpred[i] <- predvals[i]
                        }
                  prederr<-((y-cvpred)^2)
                  CV <- mean(prederr)
#   create diagonal matrix of weights from robust estimator
                  reg<-method(x,y)
                  wt<-diag(reg$weight,ncol=n)
#   find weighted avg prediction error
                  CVwt <- mean((wt%*%prederr))
                              }
            return(CV, CVwt)

}
#   The function cv1sim is the full simulation for the leave-one-out estimate
#   of prediction error.  Input the number of replicates (iter),the number of
#   of outliers (out), the magnitude of the outliers (dis), the noise to
#   generate the response N(0,sigma) and the seed so results can be duplicated
#   and the same datasets are used except factors altered.  If other data sets
#   are used like Gunst and Mason, you don't need all those parameters.
#   estimator is the regression estimator that must have at least $coef and
#   $weight for weighted avg prediction error. Note that dis is delta*sigma in
#   table 5.14, so 5delta and 5sigma means dis=25 for simulation.
cv1sim<-function(iter, out, dis, sigma,seedy,estimator)
{
            {
```

```
                        cumwin.shao <- matrix(0, nrow = 1, ncol = 5)
                        cumwin.jw <- matrix(0, nrow = 1, ncol = 5)
                        cumwin.shaowt <- matrix(0, nrow = 1, ncol = 5)
                        cumwin.jwwt <- matrix(0, nrow = 1, ncol = 5)
                    for(i in 1:iter) {
                        cat("iter ", i, " ")
                        seeder <- seedy + i
                        j<- gendatanormc(seeder, out,dis,sigma)
#   The following code removes the outliers if the standardized residuals
#   are larger than 2.5 for a fit with Simpson and Montgomery estimator.
#                       smreg <- bijs5sa(j$x, j$y)
#                       absres <- abs(smreg$residuals/smreg$scale)
#                       j$x <- j$x[absres < 2.5,    ]
#                       j$y <- j$y[absres < 2.5]
#   Find SST
                        cv0 <- sum((j$y - mean(j$y))^2)/length(j$y)
#   Get cross-validation estimates of prediction error for 1 var model
                        cv1 <- cvpress(j$x[, 1], j$y,method=estimator)
                        cv2 <- cvpress(j$x[, 1:2], j$y,method=estimator)
                        cv3 <- cvpress(j$x[, 1:3], j$y,method=estimator)
                        cv4 <- cvpress(j$x[, 1:4], j$y,method=estimator)
#                       cv5 <- cvpress(j$x[, 1:5], j$y,method=estimator)
#                       cv6 <- cvpress(j$x[, 1:6], j$y,method=estimator)
#                       cv7 <- cvpress(j$x[, 1:7], j$y,method=estimator)
#                       cv8 <- cvpress(j$x[, 1:8], j$y,method=estimator)
#                       cv9 <- cvpress(j$x[, 1:9], j$y,method=estimator)
#   Now we have the 5 contending models with 2 measures of cross validation
#   prediction error for each alternative.
                    cv <- matrix(c(cv0, cv1$CV, cv2$CV, cv3$CV, cv4$CV),nrow = 1)
                    cvwt <- matrix(c(cv0, cv1$CVwt, cv2$CVwt, cv3$CVwt,
    cv4$CVwt),nrow = 1)
#   The matrix win.shao has a 1 entry for minimum prediction error
#   otherwise it is 0.
                        win.shao <- ifelse(cv == min(cv), 1, 0)
                        cumwin.shao <- cumwin.shao + win.shao
                        win.shaowt <- ifelse(cvwt == min(cvwt), 1, 0)
                        cumwin.shaowt <- cumwin.shaowt + win.shaowt
#   win.jw finds the model that meets the change in prediction error criteria.
                        win.jw <- win.pure(cv,const=.0025)
                        win.jwwt <- win.pure(cvwt,const=.0005)
                        idx <- rep(0, 5)
                        idx[win.jw] <- 1
                        idxwt <- rep(0, 5)
                        idxwt[win.jwwt]<-1
                        cumwin.jw <- cumwin.jw + idx
                        cumwin.jwwt <- cumwin.jwwt + idxwt
                            }
                }
        return(cv, cumwin.shao, cumwin.jw,cvwt,cumwin.shaowt,cumwin.jwwt)

}
#   The function crossvald computes the K-fold and adjusted K-fold estimates
#   of prediction error.  It also returns the weighted estimates if a robust
#   estimator is used.
crossvald<-function(x, y, method = bijs5, cvmse = function(y, yhat)
mean((y - yhat)^2), K = 6)
{
        {
                set.seed(129)
                x <- as.matrix(x)
                n <- nrow(x)
                out <- NULL
                f <- ceiling(n/K)
#   Sample without replacement from a vector [1, 2, 3,..K, 1, 2,..K...]
```

```
#   (there are f repetitions of 1, 2, ...K] to identify which assessment group
#   the observation belongs.  The sample size is n.  The assesment sample
#   sizes should be close to one another because we resample without
#   replacement.
                s <- sample(rep(1:K, f), n)
                y <- as.matrix(y, ncol = 1)
                regress <- method(x, y) #  find predicted values for the model
                predvals <- y - regress$residuals
#   Overall resubstitution error is corr.  This is the initial value
#   required to compute the bias correction factor.
                corr <- cvmse(y, predvals)
                CV <- 0
                CVwt<-NULL
                xint <- matrix(cbind(1, x), nrow = n)
#   For each assessment set S.as compute predicted values
                pe<-matrix(0,ncol=1,nrow=n)
                for(i in 1:K) {
#   Select observations with index i for the assessment set
                    S.as <- c(1:n)[(s == i)]
#   The training set is all the observations to remain
                    S.tr <- c(1:n)[(s != i)]
#   Perform regression with the current training set
                    cvreg <- method(x[S.tr,  ], y[S.tr])
                    predvals <- xint %*% cvreg$coef
#   The proportion of the data in the ith assessment set
                    p.alpha <- length(S.as)/n
                    pe[S.as]<-(y[S.as]-predvals[S.as])^2
                    pred.err <- cvmse(y[S.as], predvals[S.as])
                    CV <- CV + p.alpha * pred.err
                    corr <- corr - p.alpha * cvmse(y, predvals)
                    CV.C <- CV + corr
                }
#   calculate the weigted avg prediction error for uncorrected bootstrap
                wt<-diag(regress$weight,ncol=n)
                CVwt<-mean(wt%*%pe)
        }
      return(CV, corr, CV.C, cvreg$coef,CVwt)
}
#   cvksim performs the simulations for K-fold cross validation.  It is set up
#   to output the K-fold prediction error and the weighted K-fold.  It
#   must be modified to do least squares by commenting out the 2 lines in
#   crossvald for wt and CVwt and change CVwt to CV.C in cvksim as directed.
#   It is set up for a 5 variable model, change ncol= to number of variables
#   desired and add the quantities to assignments in "cv".
cvksim<-function(iter, out, dis, sigma,seedy,estimator)
{
        {
                cumwin.shao <- matrix(0, nrow = 2, ncol = 5)
                cumwin.jw <- matrix(0, nrow = 2, ncol = 5)
                for(i in 1:iter) {
                    cat("iter ", i, " ")
                    seeder <- seedy + i
                    set.seed(seeder)
                    j <- gendatanormc(seeder, out,dis,sigma)
#   code to remove outliers first with Simpson and Montgomery estimator
#                   smreg <- bijs5sa(j$x, j$y)
#                   absres <- abs(smreg$residuals/smreg$scale)
#                   j$x <- j$x[absres < 2.5,  ]
#                   j$y <- j$y[absres < 2.5]
                    nn <- length(j$y)
                    cv0 <- sum((j$y - mean(j$y))^2)/length(j$y)
#   Get cross-validation estimates of prediction error for 1 var model
                    cv1 <- crossvald(j$x[, 1], j$y, K = 6)
```

```
#   2 variable model
                    cv2 <- crossvald(j$x[, 1:2], j$y, method=estimator, K = 6)
                    cv3 <- crossvald(j$x[, 1:3], j$y, method=estimator, K = 6)
                    cv4 <- crossvald(j$x[, 1:4], j$y, method=estimator, K = 6)
#                   cv5 <- crossvald(j$x[, 1:5], j$y, method=estimator, K = 6)
#                   cv6 <- crossvald(j$x[, 1:6], j$y, method=estimator, K = 6)
#                   cv7 <- crossvald(j$x[, 1:7], j$y, method=estimator, K = 6)
#                   cv8 <- crossvald(j$x[, 1:8], j$y, method=estimator, K = 6)
#                   cv9 <- crossvald(j$x[, 1:9], j$y, method=estimator, K = 6)
#   Now we have the 5 contending models with 4 measures of cross val error
#   for each one.  Put them in a 4 by 5 matrix to find out the best model
#   under the criterion.  Note that this run is set up to evaluate the K-fold
#   and the weighted K-fold.  Simply replace cv*$CVwt with cv*$CV.C to get
#   the bias corrected versions.  To evaluate more than 5 variables remove
#   the comments from cv# above and add those variables to "cv" matrix.
                    cv <- matrix(rbind(c(cv0, cv1$CV, cv2$CV, cv3$CV, cv4$
     CV), c(cv0, cv1$CVwt, cv2$CVwt, cv3$CVwt, cv4$CVwt)), nrow = 2)
#   The matrix winners has a 1 entry if the prediction error is lowest
#   0 otherwise.
                    win.shao <- t(apply(cv, 1, matmin))
                    cumwin.shao <- cumwin.shao + win.shao
                    winidx.jw <-apply(cv,1,win.pure,const=0.0025)
                    cv.jw <- rep(0, 5)
                    cv.jw[winidx.jw[1]] <- 1
                    cvadj.jw <- rep(0, 5)
                    cvadj.jw[winidx.jw[2]] <- 1
                    win.jw <- matrix(rbind(cv.jw, cvadj.jw), nrow = 2)
                    cumwin.jw <- cumwin.jw + win.jw
          }
     }
     return(cumwin.shao, cumwin.jw, cv)
}
#   The following are example implementing the code
date()
run3cv1<-cv1sim(50,4,10,1,130,bijs5)
run3cv1
run5bs2<-shaosimgmr(25,8,25,5,20,25,155)
run5bs2
run8cvk0025<-cvksim(50,8,50,5,130,bijs5)
run8cvk0025
date()
```